STRESS ANALYSIS OF CERAMIC TURBINE BLADES BY FINITE ELEMENT METHOD - PART I.

Lael Ray Easterling

Report Number: NPS69-78-009

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

STRESS ANALYSIS OF CERAMIC TURBINE

BLADES BY FINITE ELEMENT METHOD - PART I

by

Lael Ray Easterling

March 1978

Thesis Advisor: G. Cantin

Approved for public release; distribution unlimited.

Prepared for: Naval Sea Systems Command
Washington, D. C.
Code 0331

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Stress Analysis of Ceramic Turbine Blades by Finite Element Method - Part I		5. TYPE OF REPORT & PERIOD COVERED Mechanical Engineer & Master's Thesis; MAR 78
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lael Ray Easterling		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N0002478WR8G192
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1978
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Gas Turbine, Stress Analysis, Finite Element; Pre- and Post-Processor.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The search for more efficient gas turbine engines has led to the proposal for the replacement of metal high temperature components with ceramic components. Essential to this effort is the numerical analysis of proposed designs. This thesis report describes the model discretization of a proposed blade design, the development of pre- and post-processors for the ADINA finite element code and the initial stress analysis of the modeled blade.

Approved for public release; distribution unlimited.

Stress Analysis of Ceramic Turbine Blades by Finite Element Method - Part I

by

Lael Ray Easterling
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1968

Submitted in partial fulfillment of the requirements for the degrees of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

and

MECHANICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
March 1978

ABSTRACT

The search for more efficient gas turbine engines has led to the proposal for the replacement of metal high temperature components with ceramic components. Essential to this effort is the numerical analysis of proposed designs. This thesis report describes the model discretization of a proposed blade design, the development of pre- and post-processors for the ADINA finite element code and the initial stress analysis of the modeled blade.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to the faculty and staff of the Mechanical Engineering Department of the Naval Postgraduate School for their support and motivation throughout his work at this institution. In particular, a great debt of gratitude is owed to Dr. Gilles Cantin, Professor of Mechanical Engineering, for his guidance and friendship as instructor and thesis advisor. Thanks are also extended to the staff of the W. R. Church Computer Center of the Naval Postgraduate School for their invaluable assistance.

Finally, the author thanks his wife, Shay, and daughter, Heather, for their support and forebearance during the demanding period of his academic pursuit.

I. INTRODUCTION

The need to increase fuel efficiency of gas turbines has led to a sizable research investment into suitable materials and designs of ceramic gas turbine components to replace high temperature resistant superalloys. This replacement is envisioned to yield a threefold benefit:

1) The refractory characteristics of ceramics will allow a greater source-sink temperature differential increasing plant efficiency without penalizing overall performance by requiring increased cooling air bled off the compressor.

2) The lower specific weight of ceramics relative to the metal components to be replaced will lower plant weight thereby increasing the plant's power-to-weight ratio.

3) Ceramic constituent components are, in general, domestically available giving them a strategic edge over superalloys.

The basic problem preventing realization of these benefits is the brittle characteristic of ceramics. Since the brittle property is associated with the desirable refractory characteristic, material research and development can only be expected to yield marginally better ceramics in the near future. Any short term success will, therefore, be a result of design innovations which will compensate for the brittle nature of ceramics to allow their use for components which have formerly been engineered on the basis of use of ductile materials.

Essential to this design effort is the use of numerical techniques to model proposed designs and develop alternative criteria with the ultimate goal of minimizing stress concentrations which key catastrophic brittle failure. This thesis describes the efforts of the author to utilize the computer hardware and software available at the Naval Postgraduate School (NPS) to analyze stress distributions in gas turbine ceramic replacement components. Time constraints dictated limiting the scope of analysis to the modeling and development of pre- and post-processors for a proposed first stage ceramic turbine blade subjected to centrifugal loading.

A turbine blade design, developed by the AIRESEARCH division of GARRETT CORPORATION under a Navy managed contract for development of a one-hundred hour test gas turbine engine using ceramic components for the combustor, stator vanes, and rotor blades, was used in this work. Stress distribution analyses were centered around the "Automatic Dynamic Incremental Nonlinear Analysis" (ADINA) code developed by Dr. Klaus-Jurgen Bathe of the Massachusetts Institute of Technology [Ref. 1] as implemented on the NPS IBM 360-67 computer system. The bulk of the modeling effort was accomplished on a Hewlett-Packard 9830 desk top calculator with associated graphics equipment and the PSAP1 graphics package [Ref. 2] implemented on the NPS computer by Lt. Adrian Kibler. Material properties were those of hot-pressed Silicon-Nitride provided by AIRESEARCH. Post-processing graphics utilized a

contour plotting routine developed by Gary L. Giles of Langley Research Center [Ref. 3] and implemented on the NPS computer by Dr. Gilles Cantin of NPS and the author.

The complete analytical efforts were divided into four major categories:

1) development of a discretized model of the blade and attachment design.

2) development of a pre-processor to calculate consistent centrifugal loads.

3) modification of the ADINA code to yield a maximum of stress output locations.

4) execution of ADINA with model developed and elaboration of post-processors to help in the interpretation of results.

II. ANALYTICAL MODEL DEVELOPMENT

Developing the finite element mesh of the complex geometric shapes incorporated in the blade was a compromise between geometric accuracy, mathematical compatibility and economy of effort. Using drawings provided by AIRESEARCH [Refs. 4, 5, 6], a mathematical definition of the airfoil and the root was developed separately and then mated using twenty node bricks arranged compatibly throughout the entire assembled mesh.

A. BLADE NOMENCLATURE

Figure 1 illustrates the nomenclature used in this paper to describe the gas turbine blade model and the reference system used to define the geometry. The portion of the component aerodynamically designed to be in the gas flow is termed the airfoil. The fillet is the transition section in between the airfoil and the attachment root designed to minimize stress concentrations. The attachment root (also termed the dovetail) forms the base of the airfoil, provides radial and tangential placement of the blade, and transmits the forces developed by the gas flow to the disk. A right-hand cartesian coordinate system was used to describe the geometry. The X-axis coincided with the turbine axis-of-rotation. The Z-axis coincided with the stacking axis for the airfoil profiles. The Y-axis was then mutually perpendicular to the X- and Z-axis, with the system origin on the rotation axis.

B. GEOMETRIC DEFINITION

1. Airfoil

Reference 4 provided sixty-two perimeter points for x-y plane cross-sections at ten z-levels from z=3.1 to z=4.0 inches. Linear fairing was prescribed for machining between cross-sections in order to facilitate manufacture. The airfoil tip level of 3.874 inches and z-level 3.385 inches (level of beginning of fillet definition) were plotted and stored for use in discretizing the airfoil. Because of the linear fairing technique, intermediary points were left to the ADINA code feature of node generation for definition.

2. Fillet

The fillet geometry consisted of two radii; 0.3 inch and 0.1 inch. The locus of centers of the 0.3 inch part of the fillet was at a constant level of 3.385 inches. The arc subscribed was tangent to the linear airfoil. The locus of centers of the 0.1 inch radius arc was centered such that the arc subscribed was tangent to both the 0.3 inch radius fillet section and the base which was considered to be a 3.2 inch radius right circular cylinder about the axis of rotation. This double radius design was chosen in order to reduce the possibility of stress concentrations by approximating an elliptic geometry.

Geometric definition of the fillet was made in sixty-two, two-dimensional planes parallel to the Z-axis and containing the normal vector to each of the sixty-two

defined perimeter points at z=3.385 inches (Figure 2a). The following described algorithm was developed to define the limiting points illustrated by Figure 2b.

a. Determination of the External Normal

A second degree Lagrange interpolating polynomial was passed through three successive perimeter points at the cross-section z=3.385 inches, and the first derivative was evaluated for the middle point in order to determine the slope of the tangent vector at this point. The negative reciprocal of the derivative is the slope of the line normal to the analysis point. Taking the y-component as -1 and the x-component as the derivative (dy/dx), a normal vector was defined. This vector was then normalized to give a unit vector. The analysis point was tested to determine the desired coefficient signs in order to define the normal with the direction external to the body of the airfoil. This procedure was carried out for each perimeter point, thereby defining sixty-two normals from the airfoil.

b. Locus of 0.3 Inch Radius Centers

The coordinate system was translated and rotated to yield a working plane which was parallel to the Z-axis and contained the unit normal vector for the point being analyzed. The curvature center was adjusted in this working plane at z=3.385 inches such that the 0.3 inch arc was tangent to the linear airfoil surface. These contact points, defining the start of fillet curvature, were transformed to the original coordinate system for future use.

c. Locus of 0.1 Inch Radius Centers

The 0.1 inch radius curvature centers were defined in each of the sixty-two working planes by use of a Newton iteration scheme to adjust the center of curvature such that the subscribed arc was tangent to both the 0.3 inch arc and the top of the root. These points of tangencies were determined, transformed to the original coordinate system and stored.

d. Definition of Perimeter Point at Arbitrary Z-Level

Using the above determined points (Figure 2b), an algorithm was developed which took as input any desired z-level in the fillet region, tested for the correct definition of geometry for that level and yielded an array of sixty-two perimeter surface points for that level.

Information from the above procedure allowed the definition of a sixty-two point perimeter at any z-level of an airfoil or fillet cross-section parallel to the x-y plane.

3. Attachment Root

The attachment root geometry was defined in a vertical plane containing the stacking axis and perpendicular to the horizontal longitudinal centerline of the body of the root which has a broach angle of 23° relative to the axis of rotation (Figure 3). Nine different geometries consisting of straight lines and arcs of circles were defined by Ref. 6 in this plane for the perimeter of the root. These goemetries were mathematically defined, and the limiting point coordinates of each segment were determined. This

shape was projected into the fore and aft faces of the disk. Symmetry conditions allowed simple manipulations of coordinates in order to define the shape of the entire attachment root. An algorithm was developed to define 231 points on any desired horizontal surface.

C. ANALYSIS MESH DISCRETIZATION

Major considerations in the development of the analysis model were to define a mesh which adequately described the geometry of the blade and yielded sufficient data points for meaningful evaluation, without making the mesh so fine as to require an inordinate amount of computer time for solution. Twenty node bricks were chosen for use throughout the mesh because of their capability of accurately defining any second order geometric curve (used almost exclusively by the designers of the blade) and their isotropic sensitivity to the applied loads.

The airfoil was represented by a mesh of twelve elements arranged one deep, three high and four long (Figure 4). The top two layers defined the linearly faired section, and the bottom row defined the fillet region. Choice of nodal point coordinates were made with the assistance of plots of horizontal cross-sections of points defined by geometry definition algorithms.

Chosen for the attachment root representation were nominal three deep, four long layers of elements defining each different vertical geometric segment (Figure 5). The exact

make-up of the root mesh was controlled by the mating of the airfoil with the attachment root which caused significant distortions because the design definition of the fillet base extended beyond the limits of the top surface of the attachment root. This mis-match of geometries necessitated brute force adjustment of nodal point coordinates in order to mate the two regions into a single consistent mesh.

Once the mating process was accomplished, all the chosen nodal points and the element connectivities were card punched in the format required by ADINA. The ADINA deck was then used as input to the PSAP1 graphics package in order to de-bug the mesh. After adjustments were made to yield a visually satisfactory mesh, a trial analysis was run which brought to light numerous mathematical difficulties caused by the distortions resulting from the mating of dissimilar geometries. These were corrected, and the final mesh (Figure 6) was considered ready for analysis.

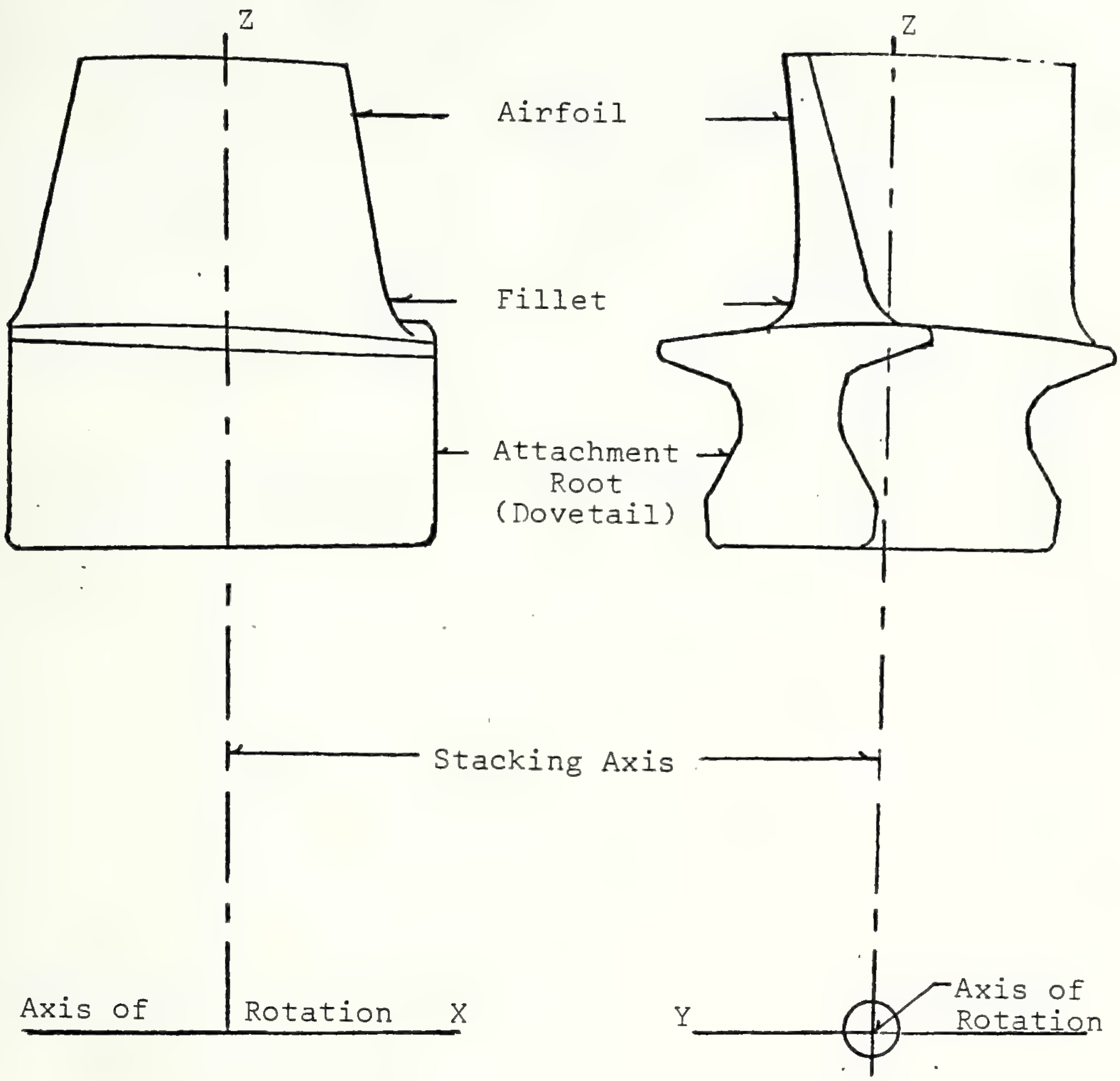


Figure 1. Illustration of Nomenclature and Reference System of Turbine Rotor Blade.

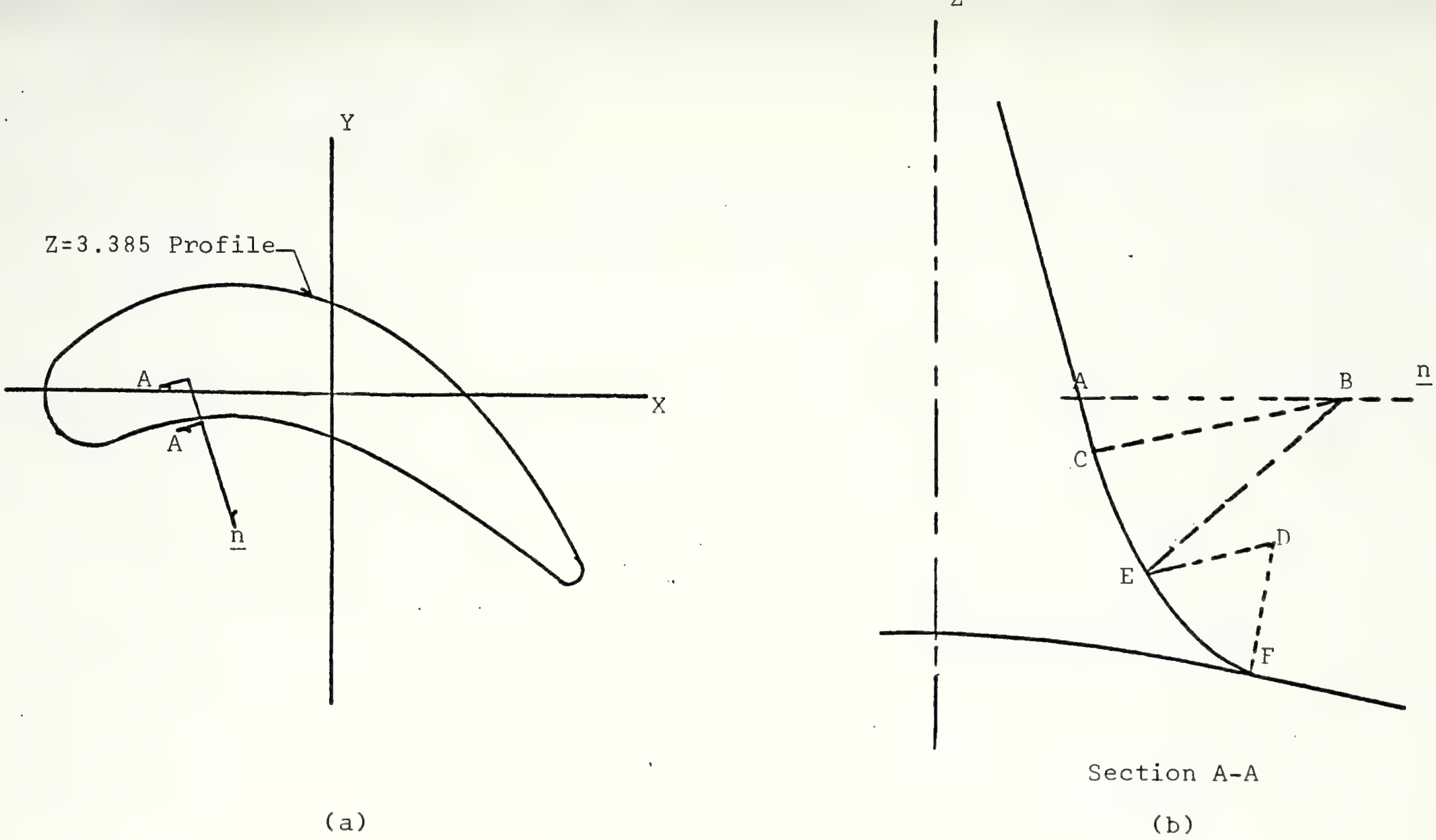


Figure 2. Illustration of Plane in Which Fillet Geometry is Defined.
Page 1 of 2

Definition of nomenclature for Figure 2

$\underline{n}$ = outward pointing normal

A = level z = 3.385 inches

B = center of 0.3 inch fillet radius curvature

C = point of tangency between linear airfoil and fillet curvature

D = center of 0.1 inch fillet radius curvature

E = point of tangency between 0.3 inch fillet curvature and 0.1 inch fillet curvature

F = point of tangency between 0.1 inch fillet curvature and attachment root surface

Figure 2. Continued, Page 2 of 2.

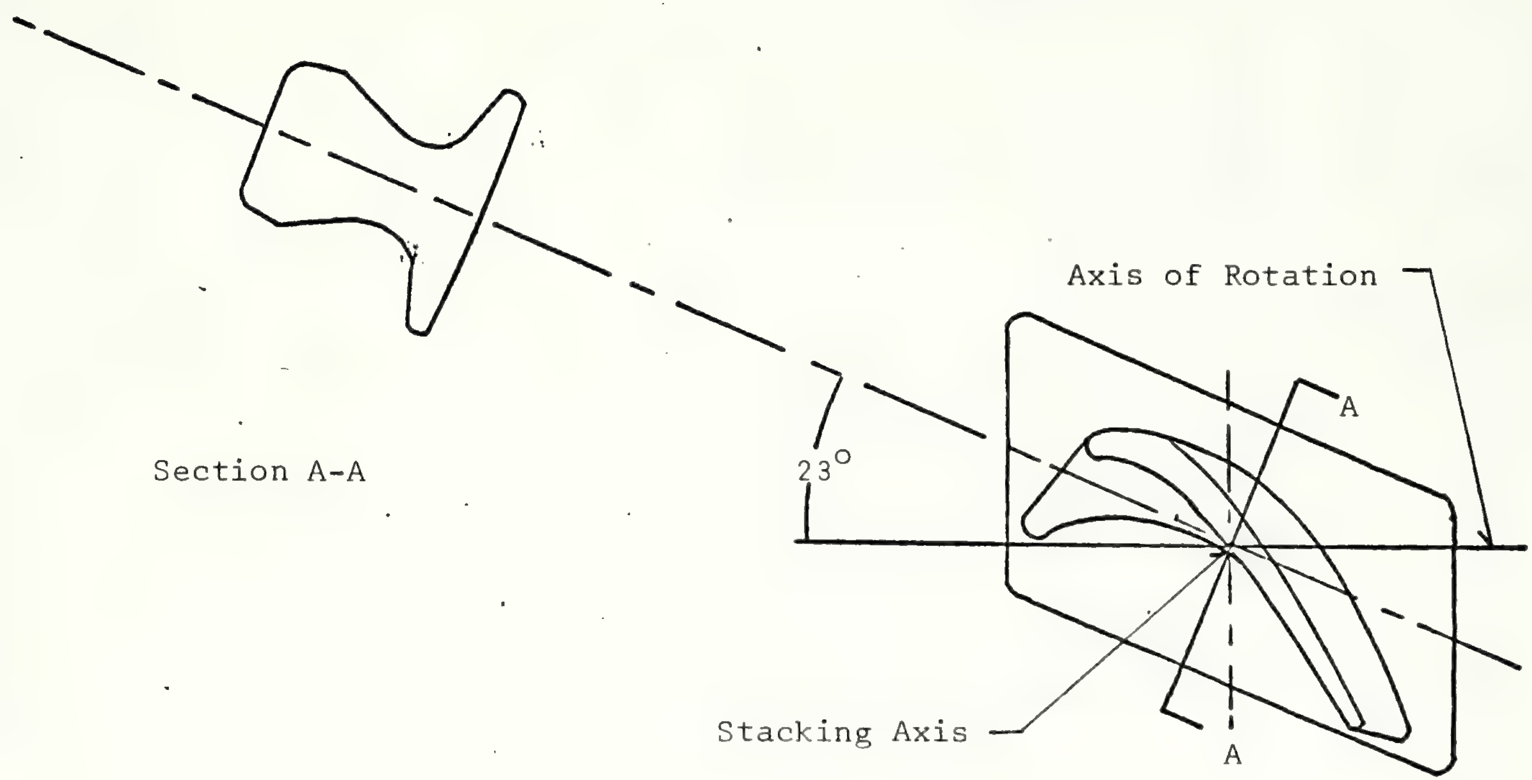


Figure 3. Illustration of Plane in Which Attachment Root Geometry is Defined.

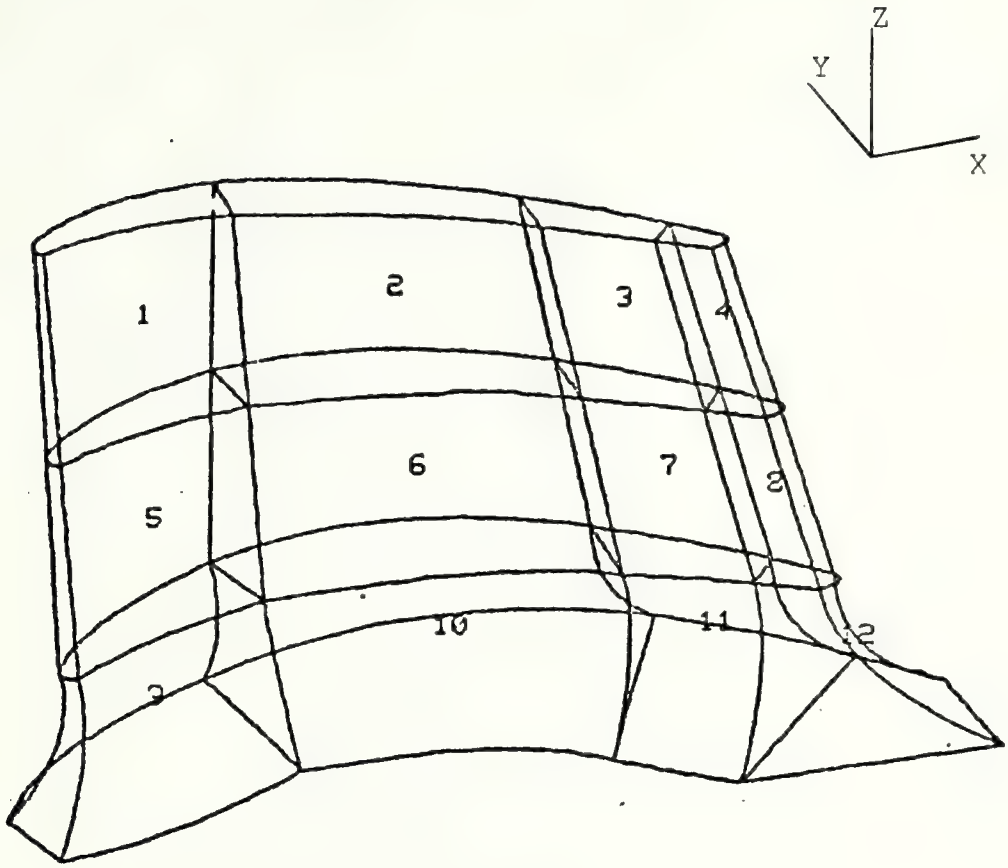
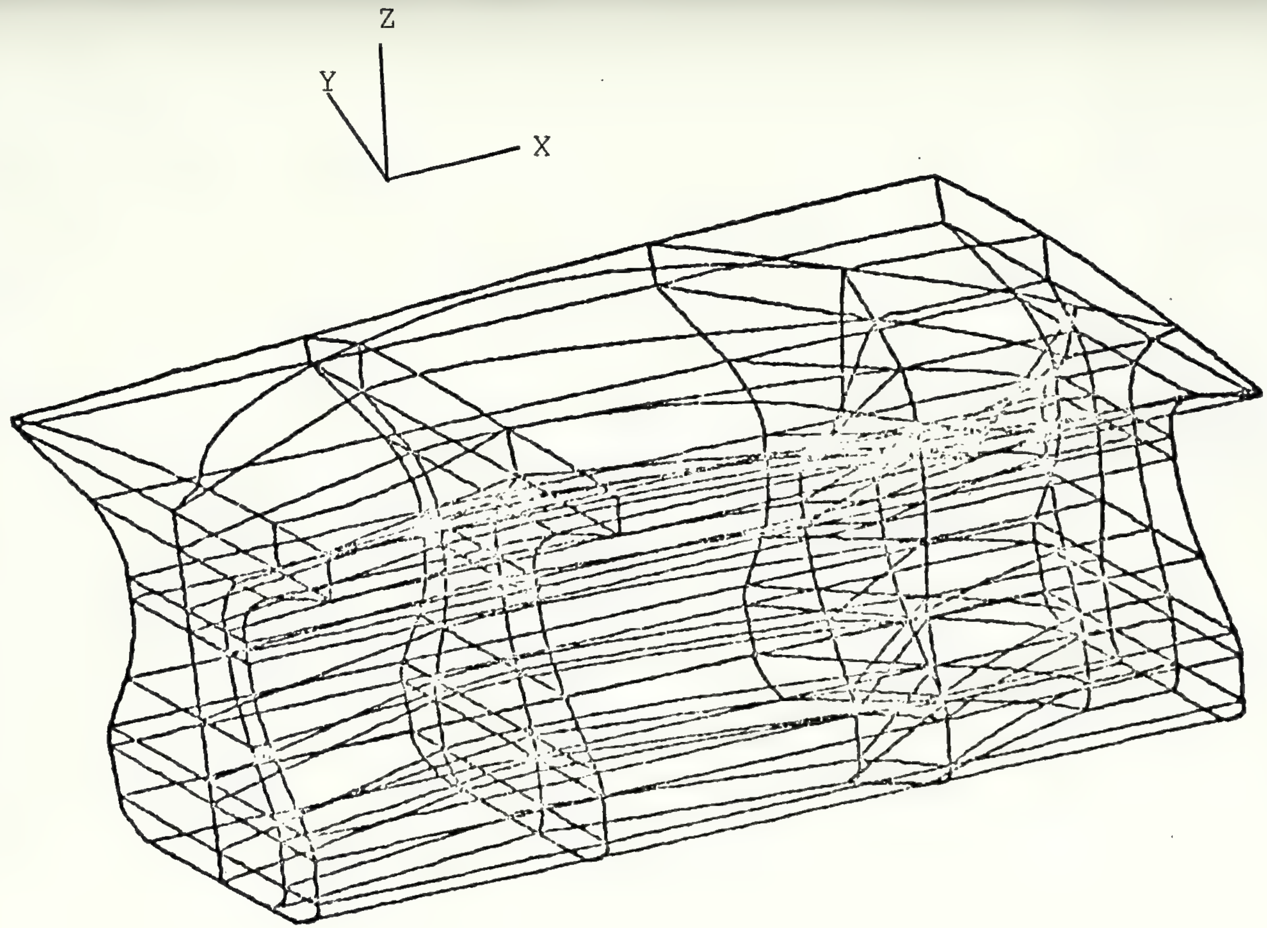
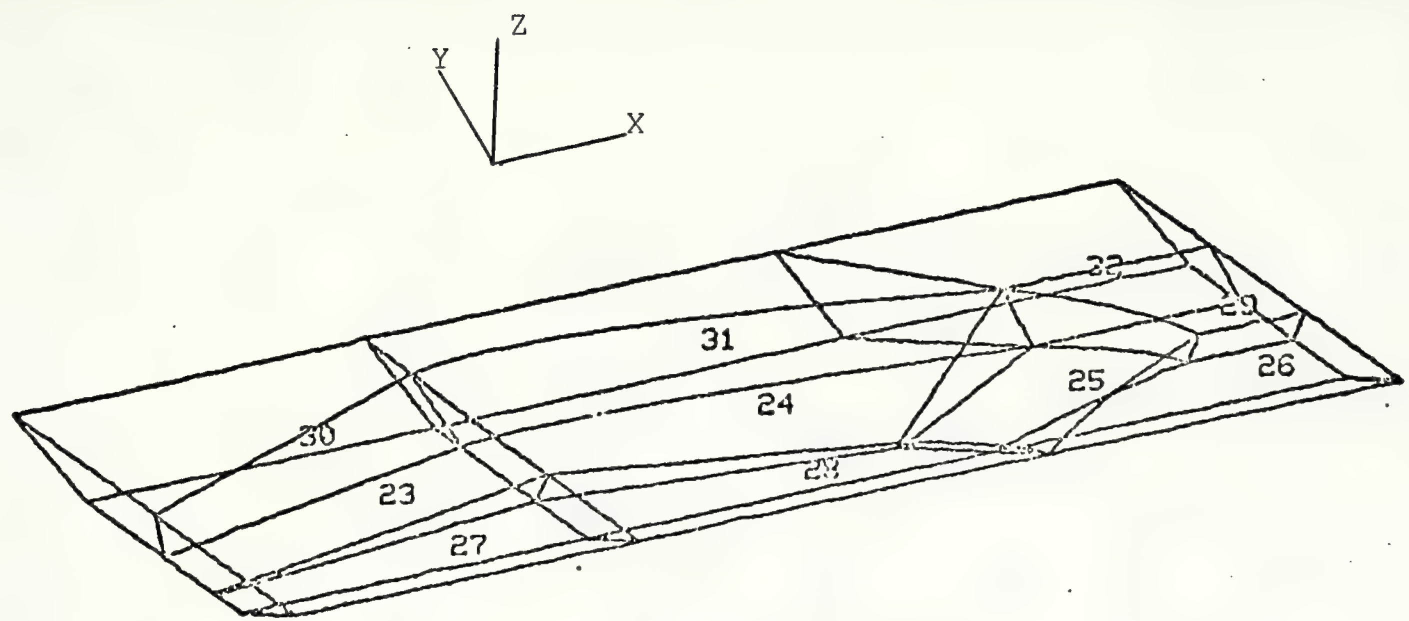


Figure 4. Airfoil Mesh, Elements 1 through 12.



(a) Elements 13 through 102

Figure 5. Attachment Root Mesh,
Page 1 of 2.



(b) Elements 23 through 32

Figure 5. Continued, Page 2 of 2.

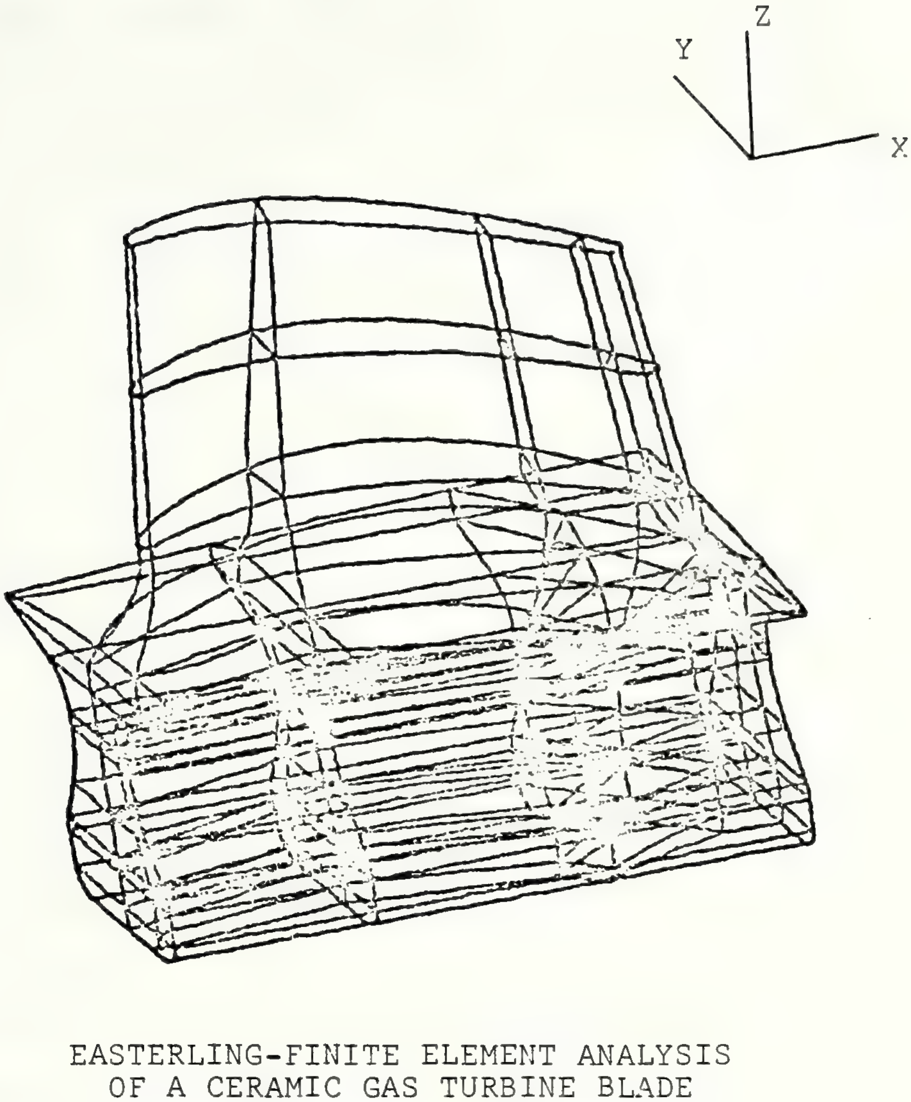


Figure 6. Completely Assembled Finite Element Analysis Mesh of Proposed Ceramic Gas Turbine Blade Design.

III. PRE-PROCESSOR DEVELOPMENT

The ADINA finite element code is a flexible, state-of-the-art code encompassing linear and non-linear, static and dynamic analyses with a choice of six types of elements and twenty material models in various combinations with element types. In this section, an explanation of the facets and capabilities of ADINA used in the blade analysis is given along with a description of the preparations that were necessary prior to execution of an analysis.

A. GENERAL DESCRIPTION OF ADINA

The brittle behavior of the hot-pressed Silicon-Nitride lends itself to the use of the linear isotropic elastic material model. Twenty node isoparametric bricks were used throughout the mesh. ADINA's use of an out-of-core solution scheme and compacted storage of matrices lends itself attractively for analysis of a large mesh which is required for the blade analysis because of the necessity of describing many different geometries.

For the chosen material model and static analysis, the equilibrium equation solved by ADINA is:

$$KU = R$$

where

K = stiffness matrix

U = displacement matrix

R = load matrix

The stiffness matrix is defined as:

$$K = \int_V B^T C B dV$$

where

B = strain-displacement transformation matrix

C = constituative stress-strain matrix

The elements of B are functions of the natural coordinates r, s, t, being derived from the isoparametric representation of displacements and the inverse of the Jacobian matrix. The integration is carried out in the natural coordinate system of reference, and dV is defined as

$$dV = \det(J)\ dr\ ds\ dt,$$

where det(J) is the determinant of the Jacobian matrix. Integration is accomplished using Gauss quadrature with the resulting matrix elements being stored in compacted form.

With the exception of gravity loading, concentrated nodal forces are the only allowed force inputs incorporated in the ADINA code implemented on the NPS computer. The rotation induced centrifugal body force needed for this analysis was obtained using an external pre-processor to be discussed in Section III-C.

In ADINA, boundary conditions constraining the structure model are imposed by eliminating global degrees of freedom of selected nodes. The real structure was constrained by a Waspaloy disk with a compliant layer interface between the ceramic contact surface and metal disk. Interpretation of

the behavior of this system of materials varies among investigators with consequent variations in modeling the boundary conditions. Without concise information on the behavior of the actual constraining mechanisms and having to limit the scope of this investigation, the tack taken was to eliminate all degrees of freedom for all nodes defining the blade contact surface.

B. MODIFICATIONS TO ADINA

The ADINA code's complex dynamic dimensioning scheme inhibits the user from making major adjustments, additions or deletions to the implemented code. Two modifications, however, were deemed important enough to implement within the code. Other problem related manipulations were implemented by external algorithms which use the ADINA input deck but do not affect the code.

Originally, ADINA came to a STOP statement when a zero or negative determinant of the Jacobian matrix was encountered. This is a valid procedure since the results of subsequent analysis would be incorrect; however, given the inexperience of this investigator and the distortions of various elements due to the requirement of producing a compatible and complete model, numerous elements were suspected of being ill-behaved, and the STOP procedure prevented efficient troubleshooting. The diagnostic output also inhibited efforts to correct model inadequacies because of a lack of identification of the offending node and element. These problems were circumvented

by deletion of the STOP statement and addition to the diagnostic the location of the offending node and element, allowing identification of all improper elements on a single trial run.

ADINA stress output consisted of three normal and three shear stresses for sixteen nodes out of a possible twenty-seven locations for each element being transmitted solely to the line-printer device. The necessity for meaningful manipulation of the massive output derived from the 682 node mesh and the desire for higher density value coverage for input into contour plotting post-processing dictated modification of the ADINA output capabilities. An additional input parameter was defined in the master control cards which allowed the user the option of requesting output values of all twenty-seven possible output locations (Figure 7). A WRITE statement was inserted to output, on a user defined file 57, all nodal stresses. Appendix A details the changes necessary to the ADINA User's Manual [Ref. 1] resulting from these modifications along with other changes from a companion investigator's contributions.

C. CENTRIFUGAL LOADING

Principal modes of loading considered for the blade were thermal, pressure and centrifugal. Given the temperature distribution, ADINA has the capacity for determining thermal gradient induced loads. External pre-processing was chosen for development of pressure and centrifugal loading. The

pressure loading pre-processor was developed by Lieutenant J. Preisel and reported in Ref. 7. A centrifugal loading algorithm was developed by this author and is described herein.

The ADINA version implemented on the Naval Postgraduate School computer accepts only concentrated nodal forces. An algorithm was developed which would calculate the nodal consistent loads equivalent to a rotation induced body force and produce an output of punched cards in a format acceptable to ADINA.

1. Mathematical Formulation

Starting with Newton's Second Law:

$$\underline{F} + \underline{I} = 0$$

$$\underline{I} = -m\,\underline{a} = -\int_V \rho\, dV\, \underline{a}$$

where

$\underline{F}$ = external forces

$\underline{I}$ = inertia forces

m = mass

ρ = mass density

$\underline{a}$ = acceleration

dV = differential volume

A vector of discretized consisten nodal forces (V_i) is desired such that

$$\langle V_i \rangle \{u_i\} = \langle I \rangle \{u\}$$

where

$\langle V_i \rangle$ = vector of consistent loads

$\{u_i\}$ = vector of discrete displacements

$\langle I \rangle$ = vector of inertia force components I_x, I_y, I_z

$\{u\}$ = vector of displacement function components $\langle u, v, w \rangle^T$

Using a right-hand cartesian coordinate system and isoparametric element definitions, displacements and coordinates are discretized by

$$\begin{aligned} \{u\} &= \langle h_i \rangle \{u_i\} \\ \{x\} &= \langle h_i \rangle \{x_i\} \end{aligned} \qquad (3)$$

where

$\langle h_i \rangle$ = vector of shape functions = f(r,s,t)

$\{x\}$ = continuum coordinates $\langle x,y,z \rangle^T$

$\{x_i\}$ = discrete nodal coordinates

The acceleration matrix is defined by

$$\underline{a} = \underline{w} \times (\underline{w} \times \underline{r}) \qquad (4)$$

where

$\underline{w}$ = angular velocity vector = $w_x\underline{i} + w_y\underline{j} + w_z\underline{k}$

$\underline{r}$ = position vector = $x\underline{i} + y\underline{j} + z\underline{k}$

$\underline{a}$ = acceleration vector = $a_x\underline{i} + a_y\underline{j} + a_z\underline{k}$

After vector algebra manipulation, one derives

$$\{a\} = [A]\,\{x^*\} \qquad (5)$$

where

$$A = \begin{bmatrix} -(w_y^2+w_z^2) & w_x w_y & w_x w_z \\ w_x w_y & -(w_z^2+w_x^2) & w_y w_z \\ w_x w_z & w_y w_z & -(w_x^2+w_y^2) \end{bmatrix}$$

$$\{a\} = \langle a_x, a_y, a_z \rangle^T$$

$$\{x^*\} = \langle x, y, z \rangle^T$$

Discretizing equations (1) and (5), using equation (3), and substituting into equation (2), one has

$$\langle V_i \rangle \{u_i\} = \rho \int \langle x_i \rangle \{h_i\} [A] \langle h_i \rangle \, dVol \, \{u_i\}$$

where

$$dVol = \det J \, dr \, ds \, dt$$

Simplifying and rearranging yield the following equation which was used to program the computer algorithm for development of consistent loads:

$$\{V_i\} = \rho \int \{h_i\}[A]\langle h_i \rangle \{x_i\} \quad dVol$$

2. Computer Algorithm for Program Centrifugal Load

The centrifugal load algorithm is generalized for use by any ADINA three-dimensional brick element mesh defined by eight to twenty nodes per element. A dynamic dimensioning scheme is used in order to conserve core and simplify changes dictated by the size of the user's mesh.

a. Input

Angular velocity about the three cartesian axes, mass density and an ADINA input deck are the required input for calculations in Program Centrifugal Load. In addition, several housekeeping parameters are required as defined in Appendix C.

The ADINA input deck provides node coordinates and mesh connectivity. By using the subroutine RDADIN, duplication of the considerable effort to punch and debug a large mesh is obviated. This subroutine should prove very useful for future users of ADINA who wish to manipulate mesh values.

b. Calculation of Consistent Nodal Loads

The user can select two to six Gauss point integration in the calculations of the consistent loads. Shape functions were derived from Ref. 8, which are the same used in ADINA. After calculation of loads for each element, the contributions to each node are summed in each of the three cartesian coordinate directions to yield the desired output of concentrated nodal forces. ADINA has the capability of summing concentrated nodal forces in a common direction for a given node; therefore, loads other than centrifugally induced may be input by the user without necessitating modification to Program Centrifugal Load output.

c. Output

Four options allow the user to restrict the output to what is desired for the particular problem under investigation:

(1) ICHK1. This flag allows the option for printout of the nodal coordinate data and mesh connectivity. Since this data is also output by use of the ADINA code and PSAP1, the user would normally not desire this option from Program Centrifugal Load.

(2) ICHK2. This parameter controls the option for controlling the output of consistent loads calculated for each element. The information may be useful in comparing the contributions of various elements.

(3) ICHK3. The information required by ADINA is controlled by this parameter which allows printout of the totaled consistent loads for each node for the three coordinate directions.

(4) ICHK4. This parameter gives the user the option of not punching the data cards as may be the choice for a first run.

Other output items include the total force in each of the three coordinate directions and the total number of load cards punched. The latter value is required by ADINA as parameter NLOAD on the Load Control Card described in Section IV of Ref. 1.

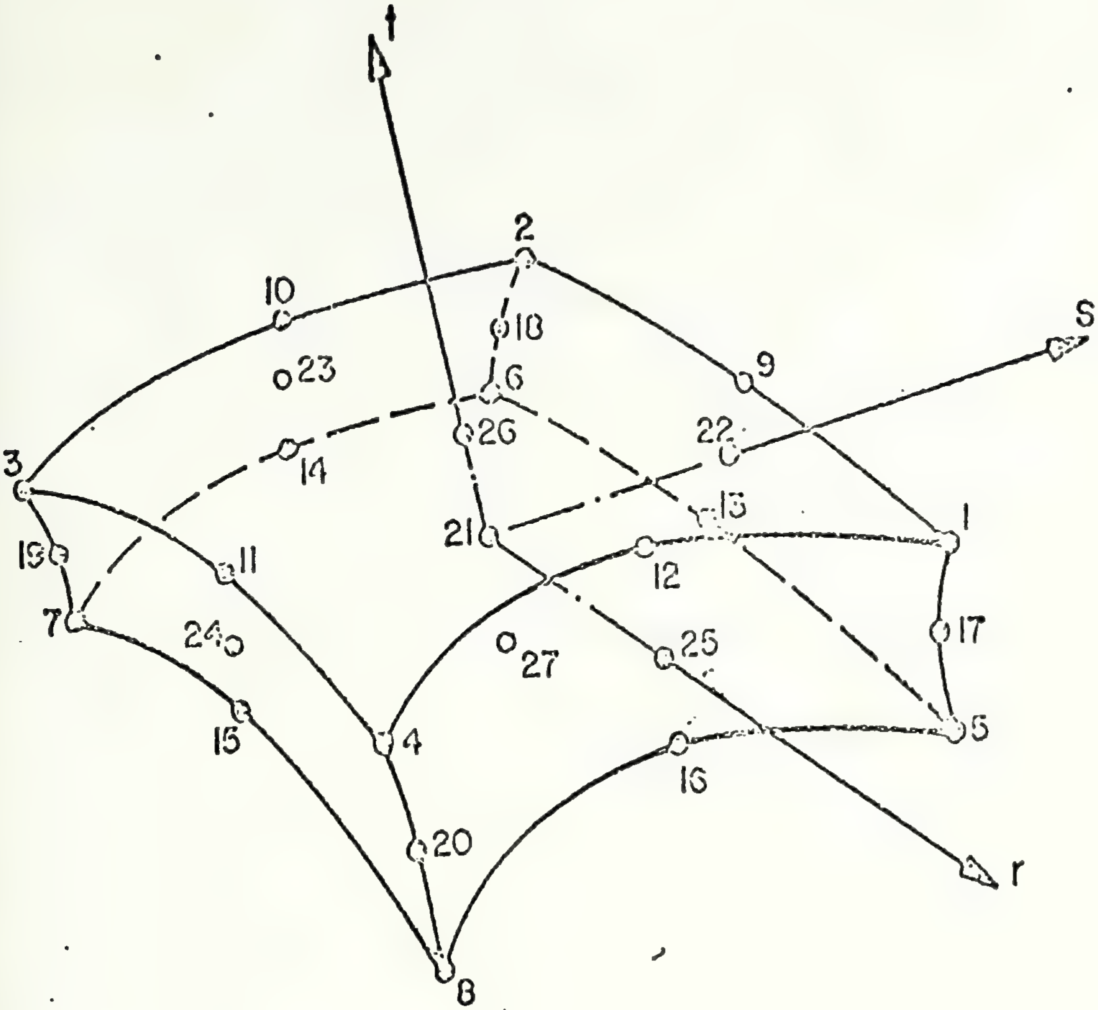


Figure 7. Stress Output Locations for ADINA
Linear Static Analysis Results.

IV. PROBLEM SOLUTION

The ADINA problem solution was executed for two, three and four Gauss points used for development of the stiffness matrix. Static linear analysis was chosen for the solution method, and a material model for isotropic behavior was used. A Young's modulus of 4.0×10^7 psi was used, corresponding to the approximate value for hot-pressed Silicon-Nitride at 2500°F. Poissons ratio was chosen to be 0.33. Loading was accomplished using concentrated nodal forces with values obtained from the centrifugal load pre-processor for an angular velocity of 42,000 rpm about the X-axis. Table 1 summarizes the execution times for the three analyses.

OPERATION	ORDER OF INTEGRATION		
	2	3	4
INPUT PHASE	22.74	20.83	24.52
MATRICES ASSEMBLAGE	717.19	1437.83	3305.98
TRIANGULARIZATION OF STIFFNESS MATRIX	637.36	663.63	705.98
STEP BY STEP SOLUTION	110.78	111.29	110.11
TOTAL SOLUTION TIME	1549.57	2234.96	4148.07

TABLE 1. ADINA PROBLEM SOLUTION TIMES (seconds)

V. POST-PROCESSOR DEVELOPMENT

Static linear analysis using the modified version of ADINA yields three orthogonal displacements for each input node and three normal and three shear stresses at each of twenty-seven locations in a twenty node brick. The value of the stress components calculated at a given node will, in general, differ from element to element. Experience has shown that the straight average of these values give the best estimate of the real stress at that node [Ref. 8]. For problems involving brittle material, the principal stresses (particularly the maximum principal stresses) are the primary values of concern in predicting failure of the modeled component. These factors necessitated the development of post-processors to manipulate the ADINA output into the values required for analysis of results. The consequent output of these post-processors was still unmanageable from the standpoint of visual perception of the relation of the results with model geometry. Additional post-processors were, therefore, developed to yield the coordinates and connectivity of a twenty-seven node brick mesh and output of punched cards in a format applicable to the contour plotting code implemented on the Naval Postgraduate School computer.

A. PROGRAM KCONT

Prior to data manipulation, the coordinates of all nodes of the twenty-seven node brick mesh and the connectivity must be known. Program KCONT was developed to accomplish this task by using the basic isoparametric finite element model equalities:

$$x = \sum_{i=1}^{n} h_i \, x_i$$

$$y = \sum_{i=1}^{n} h_i \, y_i$$

$$z = \sum_{i=1}^{n} h_i \, z_i$$

where

x, y, z = global coordinates of any point within the element

x_i, y_i, z_i = global coordinates of the nodes defining the element

h_i = the array of shape factors as the function of the natural coordinates r, s, t.

Program KCONT first reads an ADINA input deck. It takes the input node coordinates and mesh connectivity data and computes the coordinates of the additional nodes. The twenty-seven node brick mesh connectivity is assembled and is output along with the coordinates of all nodes to the line printer and to the user defined storage files in accordance with the instructions in Appendix D.

B. PROGRAM STRESS

Program Stress reads the file defined by the ADINA code user for storage of stress results and the files storing the total connectivity and coordinate information, averages all values for each given node and determines the principal stresses from the average values. The output consists of storage of the results on user defined files and a line printer presentation. Refer to Appendix E for instructions on using Program Stress.

C. PROGRAM CONTOUR PLOT DATA

Data required by the two-dimensional iso-line contour plotting code consists of housekeeping information in the form of defined NAMELISTS, coordinate data of nodes on the analysis plane, four-node two-dimensional connectivity and input values of information to be plotted. Program Contour Plot Data was designed to define the desired analysis plane(s) and yield a punched data deck of coordinates and plotting value information. A graphics generated plot of node points and global numbers is also generated to assist the user in the formulation of the connectivity. The connectivity information must then be punched on data cards by the user and inserted into the appropriate data deck location.

Analysis planes can be defined with any of three options. The user can simply input the nodes to be used on data cards in the format 16I5. Most models, however, have surfaces of interest which are parallel to one of the three orthogonal

planes defined by the axes of the right-handed cartesian coordinate systems. Should this be the case, the user may define bounding values of a coordinate for testing the mesh for the desired nodes defining the analysis plane. If the user inputs both bounding values equal, an equality test is made in order to define the desired plane.

The program as written takes the array of plotting values from the file containing the principal stresses established by Program Stress and uses the maximum principal stress array (variable SIGMX) for plotting contours. A user may redefine the read statement (line CTRP2470 of Appendix F) in order to input the desired values for a particular problem, filling the array SIGMX with the desired plotting data. Coordinate values are read from file 58 established by Program KCONT.

The resulting punched deck of cards requires the input of connectivity and NAMELISTS &OPTION and &PICT in locations designated by Ref. 3.

D. PROGRAM CONTOUR PLOT

The contour plotting routine chosen to display stress results was developed by Gary L. Giles of the Langley Research Center for use with a variety of graphic systems including the CALCOMP plotter installed at the Naval Postgraduate School. Unfortunately, the NPS graphics software package rotates the plotting axis 90° clockwise from the original software conception of plot orientation. In order

to achieve optimum utilization of the plotting surface, the decision was made to modify the contour plot routine to utilize the NPS graphics package but yield a plot oriented with the horizontal plotting axis along the length of the paper roll as originally conceived by the CALCOMP manufacturer. In general this requires inputting to the drawing routines of NPS the negative of the desired vertical coordinate in the calling location for the x-coordinate and the positive horizontal coordinate in the calling position of the y-coordinate.

The plotting origin for the analysis plane is chosen to be the geometric center. In order to place the plot properly on the plotting surface for any set of coordinates, parameters PXORGN and PYORGN were added to NAMELIST &OPTION with default values of 0.0. Inputting the user's coordinates for the center of his analysis plane properly centers the plot on the plotting surface.

VI. RESULTS OF ADINA STRESS ANALYSIS

Stress results were analyzed using a program developed to compare the results of the two, three and four point Gauss integration analyses and plotting the iso-maximum principal stress contours for various chosen planes of analysis. Severe surface tensile stress concentrations were observed in the region immediately above the ceramic-disk contact region.

A. COMPARISON OF MAXIMUM PRINCIPAL STRESSES

Formulation of the stiffness matrix using two-point integration resulted in higher stresses caused by a lower order in integration, creating a more flexible system which concurs with finite element research [Ref. 8]. Some investigators have found that the results of reduced order integration reflect more accurately the actual component stresses because the strict mathematical development of the finite element method creates a system which is stiffer than the actual component. Insufficient experimental data, uncertainty of boundary conditions and the lack of an adequate convergence study prevent a statement of qualitative opinion of the actual stresses involved for the blade analysis; however, noting the failure rate of the few specimens which have been tested, it is probable that tensile stresses in the dovetail are higher than predicted.

Table 2 illustrates the differences between the three analyses by comparing the greatest maximum principal stress

encountered, the average of maximum principal stresses and the Euclidean norm of the differences between results for different integration order analysis.

	ORDER OF INTEGRATION		
	2	3	4
MAXIMUM PRINCIPAL STRESS	52862.39	46671.73	4674.73
AVERAGE MAX PRINCIPAL STRESS	6198.79	6100.42	6093.98
	INTEGRATION ORDER ANALYSES COMPARED		
	2-3	2-4	3-4
EUCLIDEAN NORM	122.2901	122.6712	4.7777

TABLE 2. COMPARISON OF RESULTS OF ANALYSES USING TWO, THREE AND FOUR POINT INTEGRATION (psi)

The maximum value occurred at node 394 which is located immediately above the contact region at level 2.9029 (Figure 8). Results from the three and four order of integration analyses showed little difference, indicating that the order of integration necessary for the "exact" evaluation of the stiffness matrix elements is being approached.

B. CONTOUR PLOT ANALYSIS

Thirty-four analysis planes were defined for contour plots. Presented herein are eight of these planes which show the regions of highest stress concentrations and the distribution of stresses in the airfoil. Figure 9 illustrates the relative position of the six plots of analysis

planes in the attachment root. The level number refers to the z-coordinate value of the nodes on that plane. The figures presented consist of an element plot, illustrating the coverage of input values to the contour plotting code, and the iso-stress plots from the second and fourth order integration analyses.

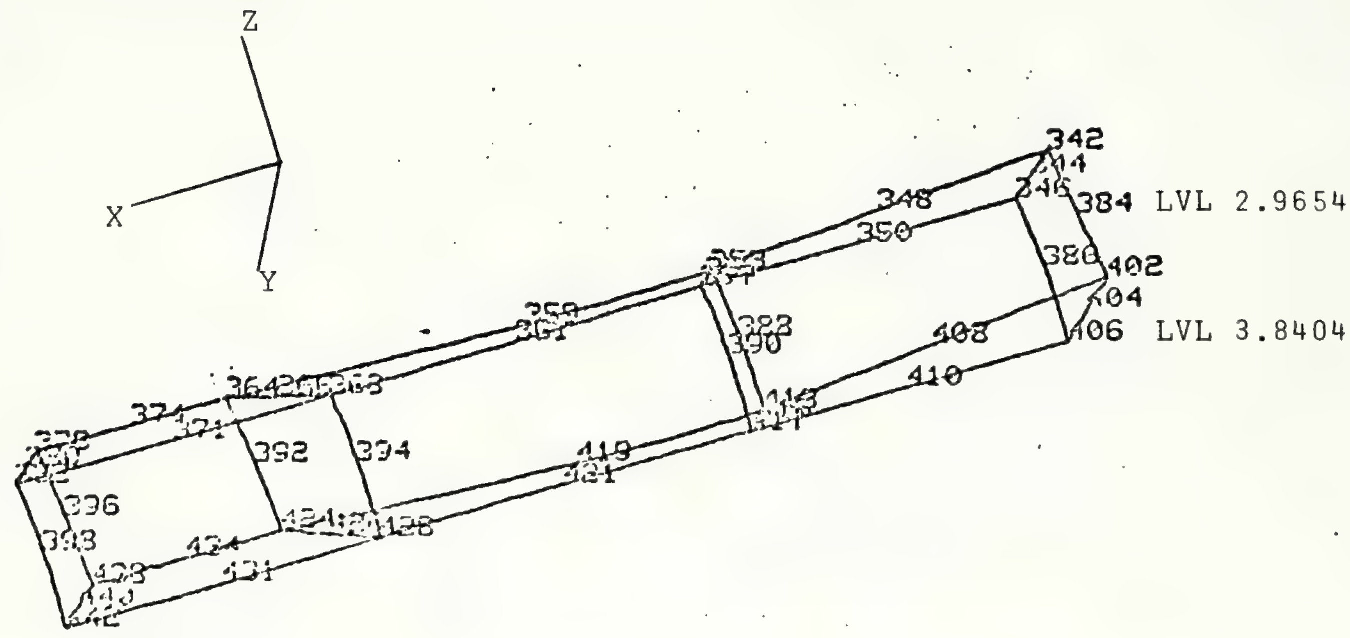
Figure 10 shows severe surface stress concentrations on both pressure and suction sides of the dovetail in the region immediately above the blade-disk contact surface. This result is consistent with other analyses of blades of similar design regardless of the boundary conditions used and loading scheme, indicating the stress distribution is principally a function of geometry. The orders of magnitude of the stress concentrations are fifty ksi for the two-point integration plot and forty-five ksi for the four-point integration plot.

Figures 11 and 12 present another view of the vertical stress distribution in dovetail. Noteworthy information from these plots is that the stress concentration is generally uniform along the length of the attachment root.

Figures 13 through 15 are plots of three horizontal surfaces in the region of high stress concentration in the dovetail.

Another region of concern in the design of a ceramic turbine blade was the fillet area of the airfoil. Figures 16 and 17 illustrate the stress distributions in the airfoil as viewed from the pressure and suction sides. Some stress

concentration does occur at the leading and trailing edges at the mid-fillet height; however, the values are minor compared to magnitudes obtained in the attachment root and relative to the strength of the material. Test results to date also indicate the adequacy of the fillet design as no failures have originated in this region except for impact initiated failure caused by flying debris upon failure of other blades.



EASTERLING-FINITE ELEMENT ANALYSIS OF A CERAMIC GAS TURBINE BLADE

Figure 8. Region of Highest Stress Concentration Elements 60, 61 and 62.

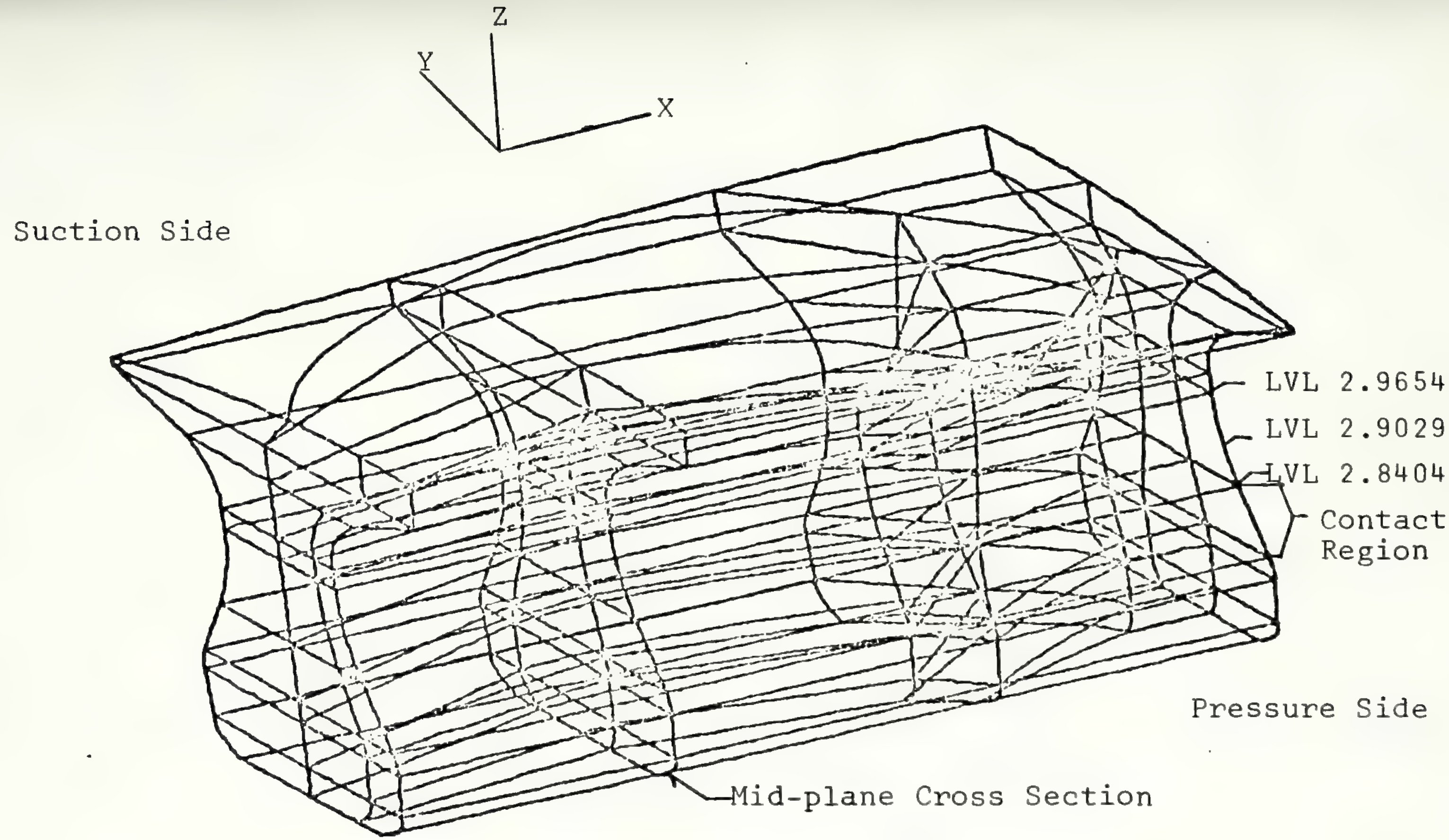
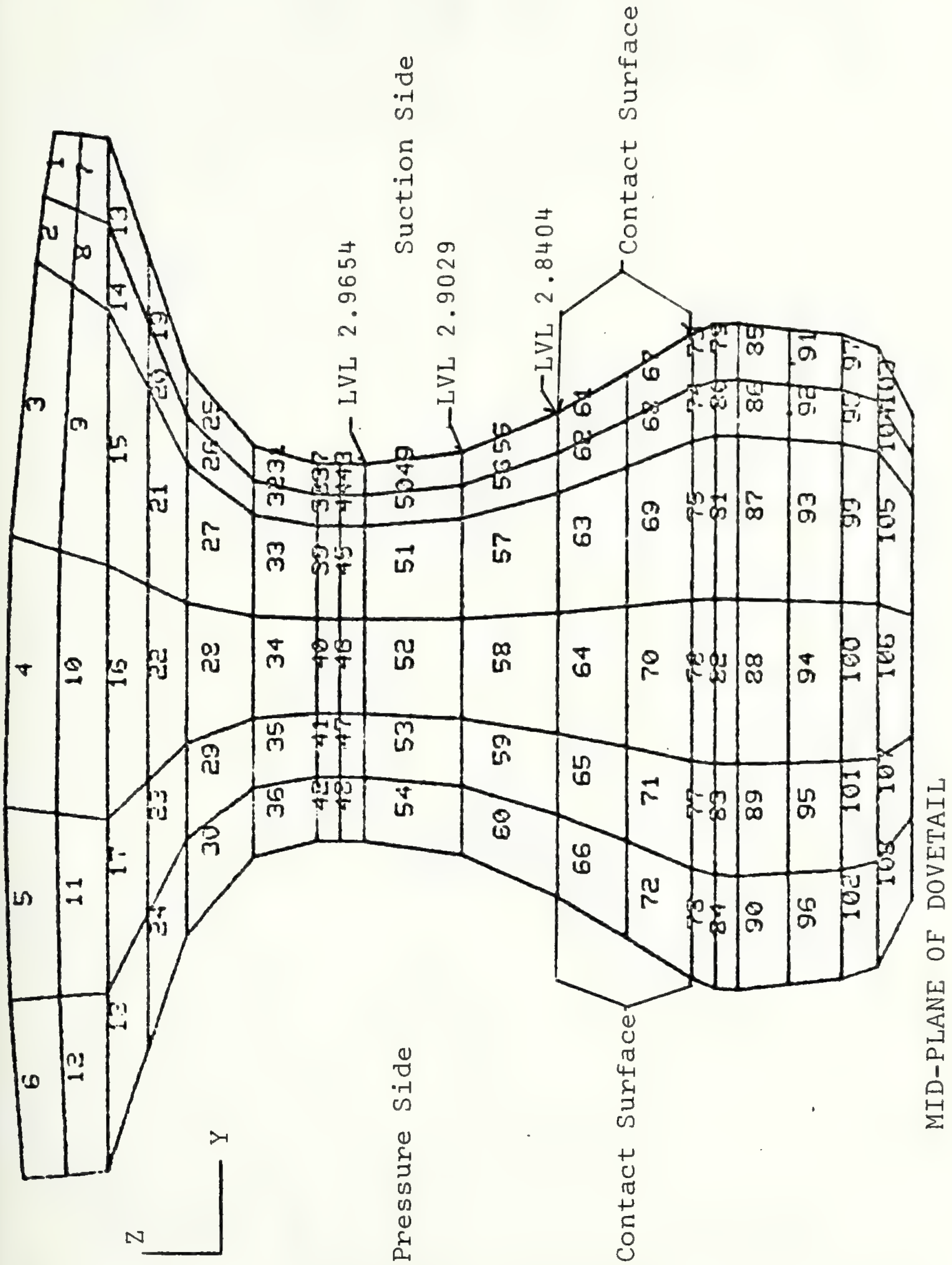
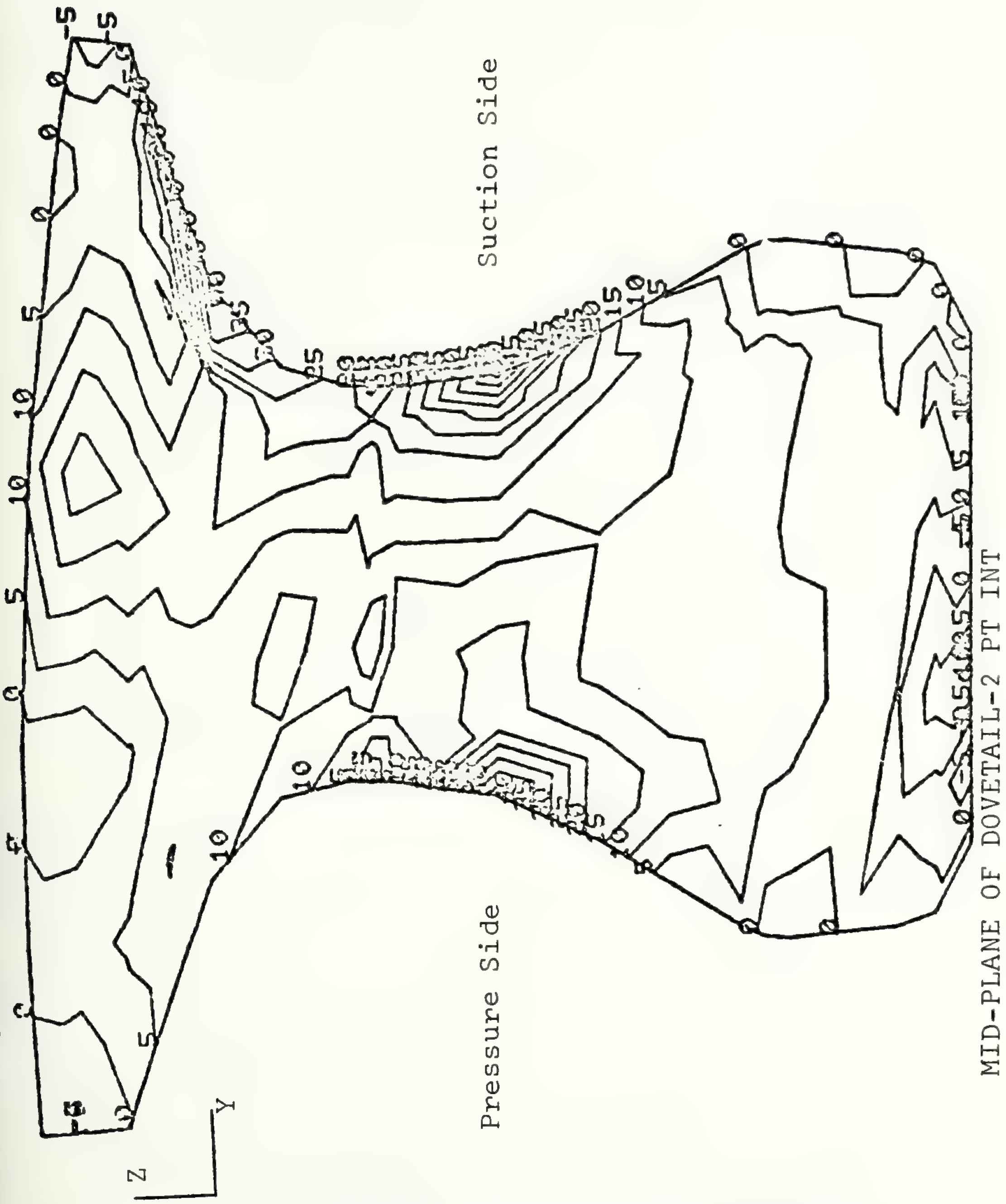


Figure 9. Attachment Root Finite Element Mesh.



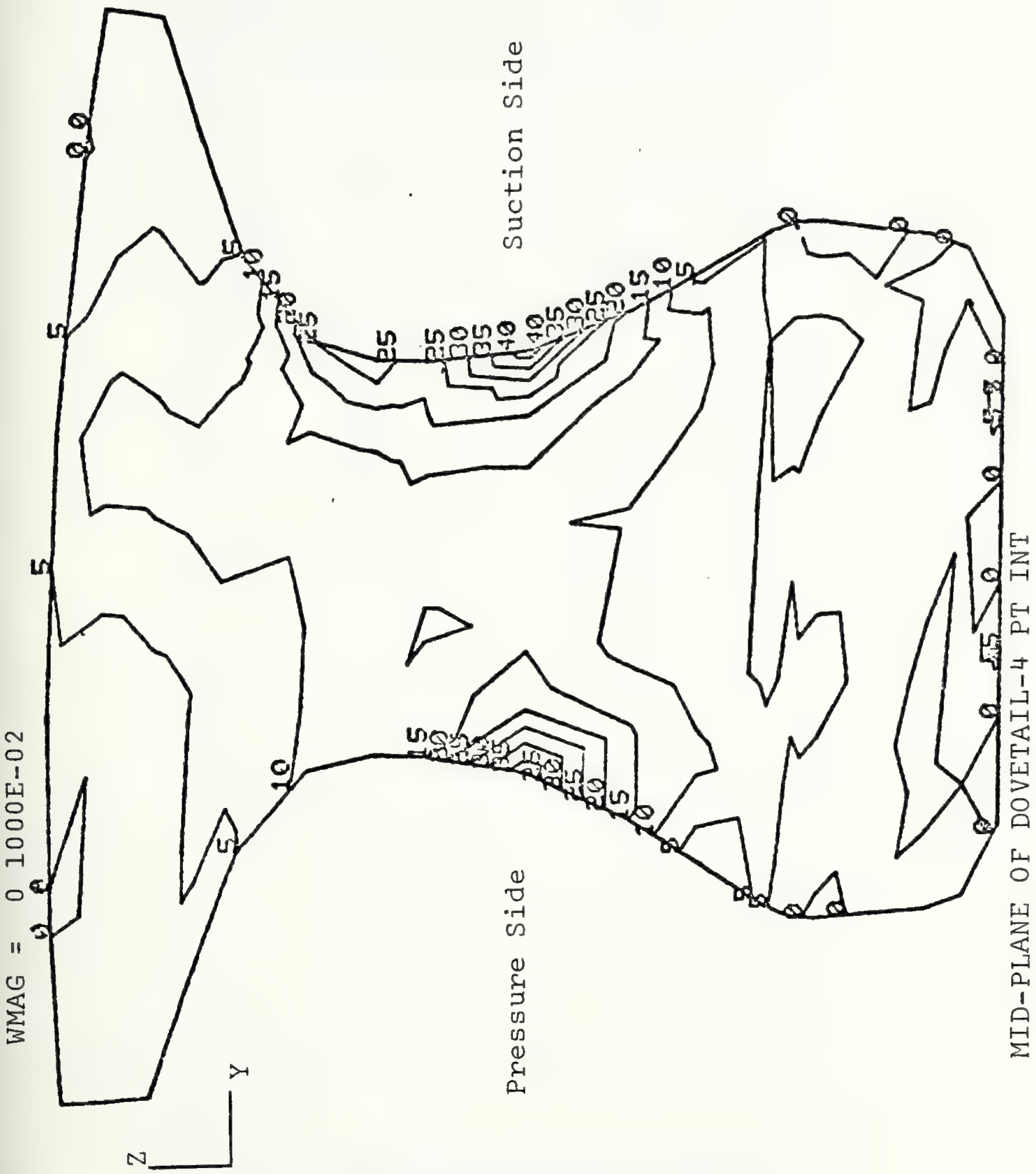
(a) 2-D Element Arrangement

Figure 10. Mid-Plane Dovetail Cross-Section, Page 1 of 3.



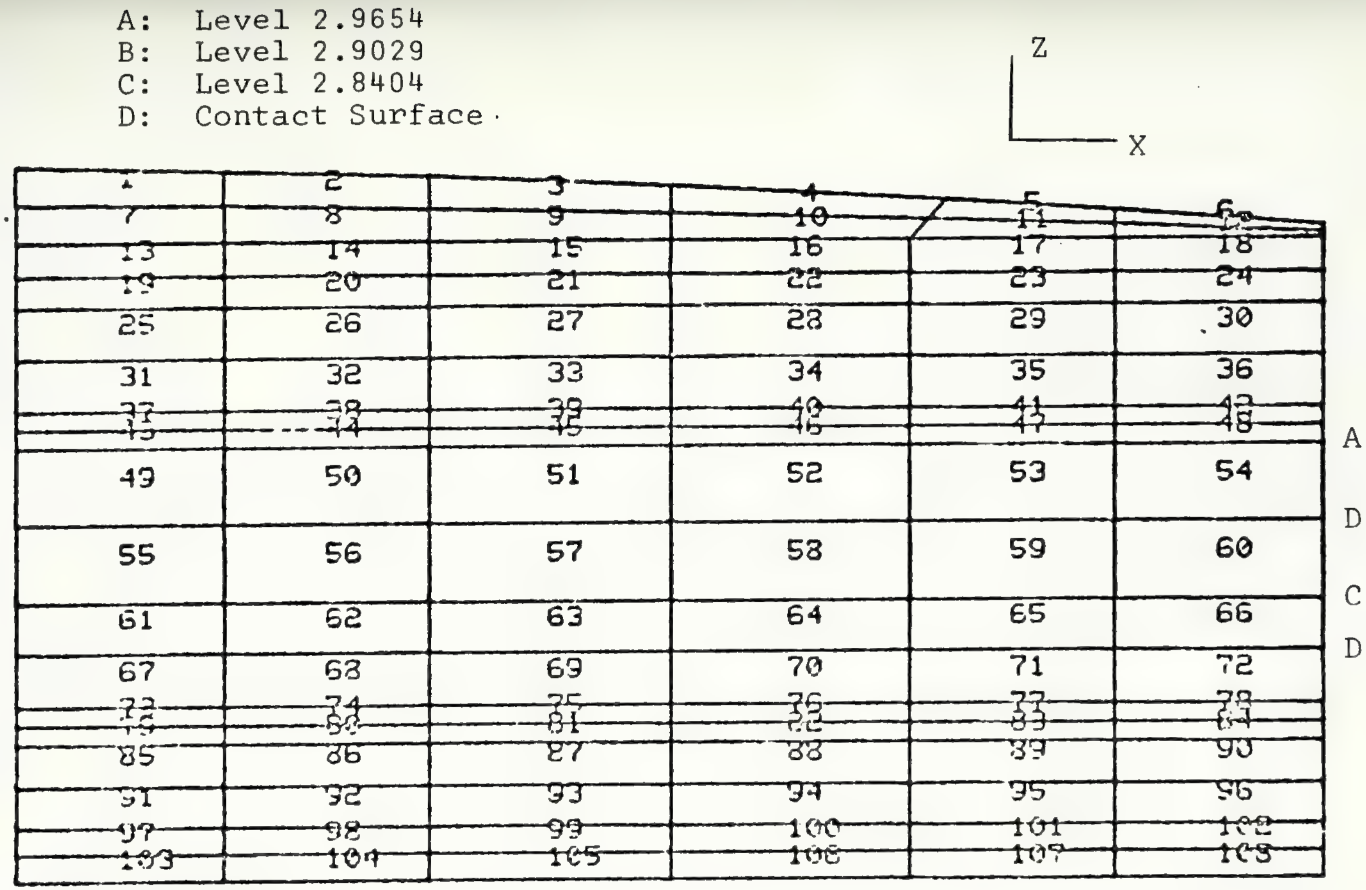
(a) Maximum Principal Stress Iso-
lines for 2-pt Order Integration (ksi)

Figure 10. Continued, Page 2 of 3.



(b) Maximum Principal Stress Iso-
lines for 4-pt Order Integration (ksi)

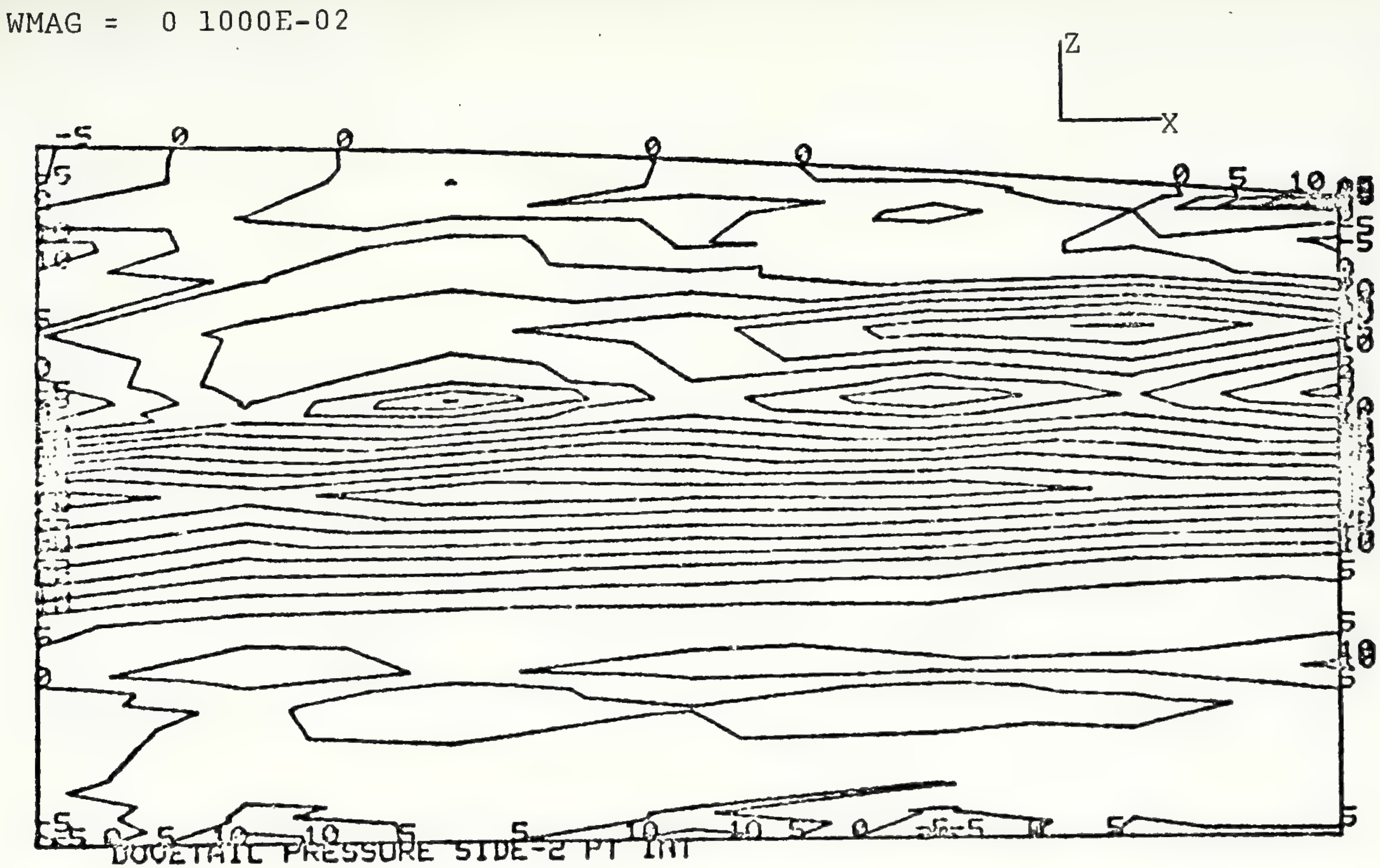
Figure 10. Continued, Page 3 of 3.



DOVETAIL PRESSURE SIDE

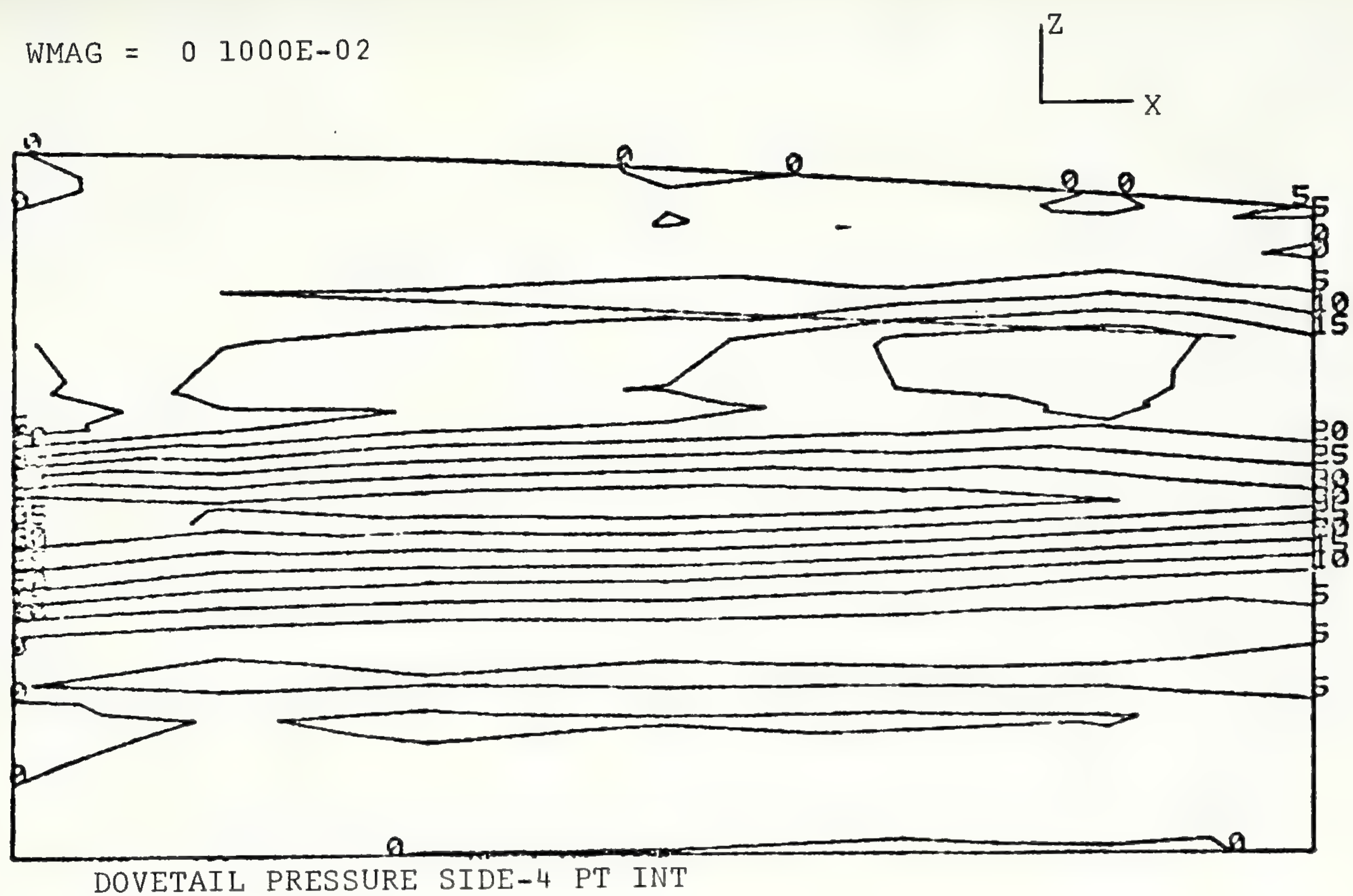
(a) 2-D Element Arrangement

Figure 11. Dovetail Pressure Side, Page 1 of 3.



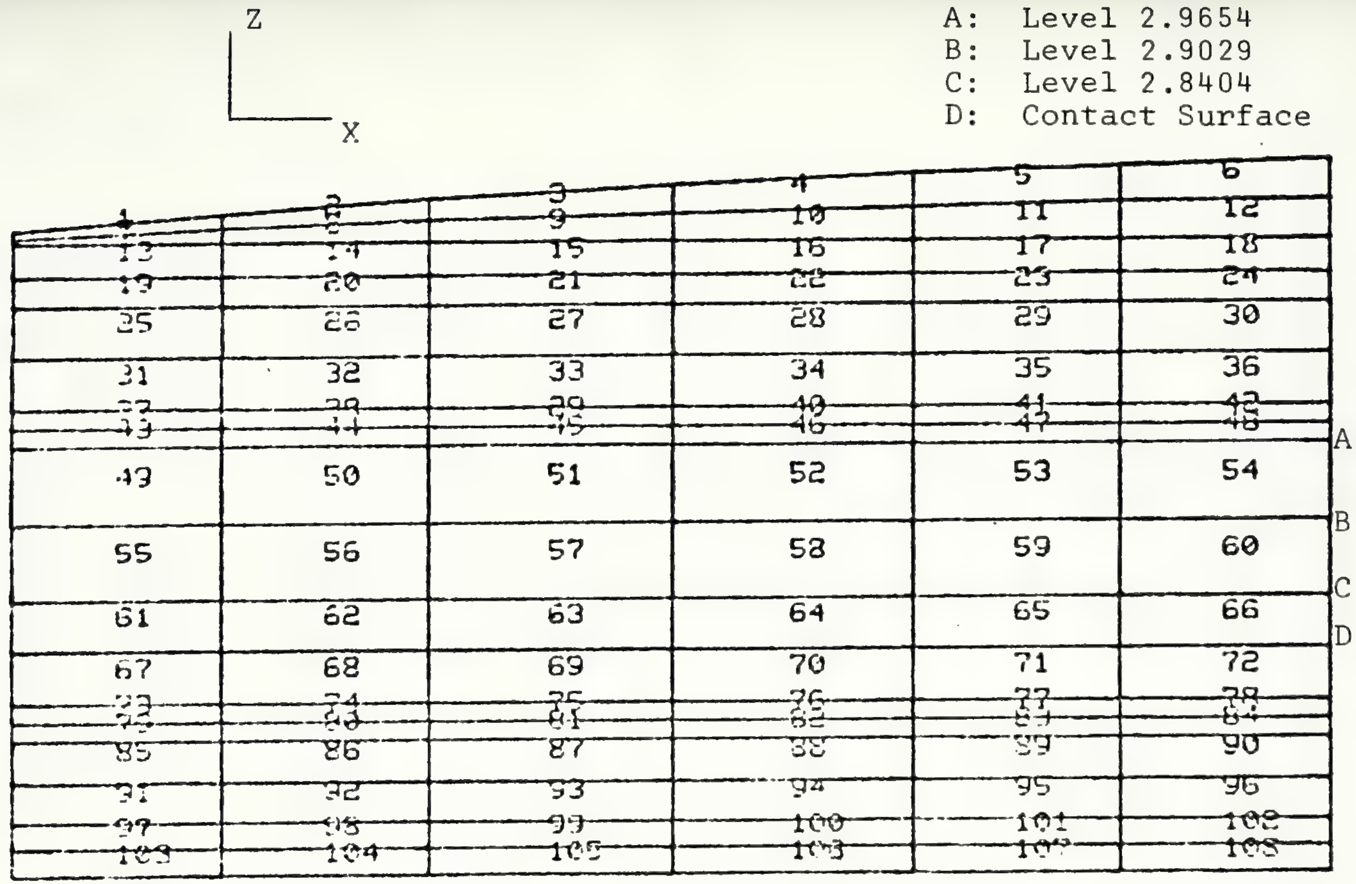
(b) Maximum Principal Stress Iso-lines
for 2-pt Order Integration (ksi)

Figure 11. Continued, Page 2 of 3.



(c) Maximum Principal Stress Iso-lines for 4-pt Integration Order (ksi)

Figure 11. Continued, Page 3 of 3.

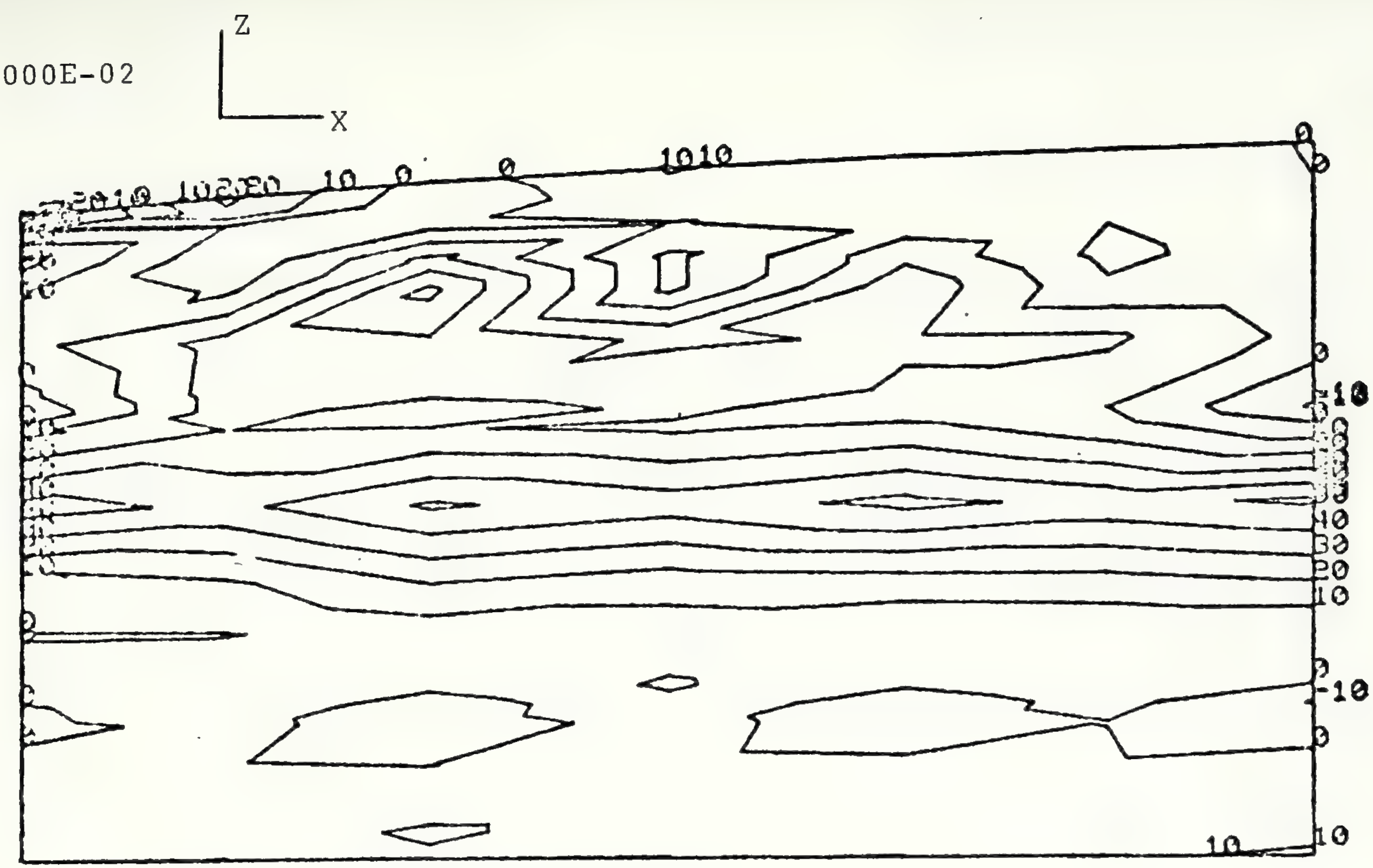


DOVETAIL SUCTION SIDE

(a) 2-D Element Arrangement

Figure 12. Dovetail Suction Side, Page 1 of 3.

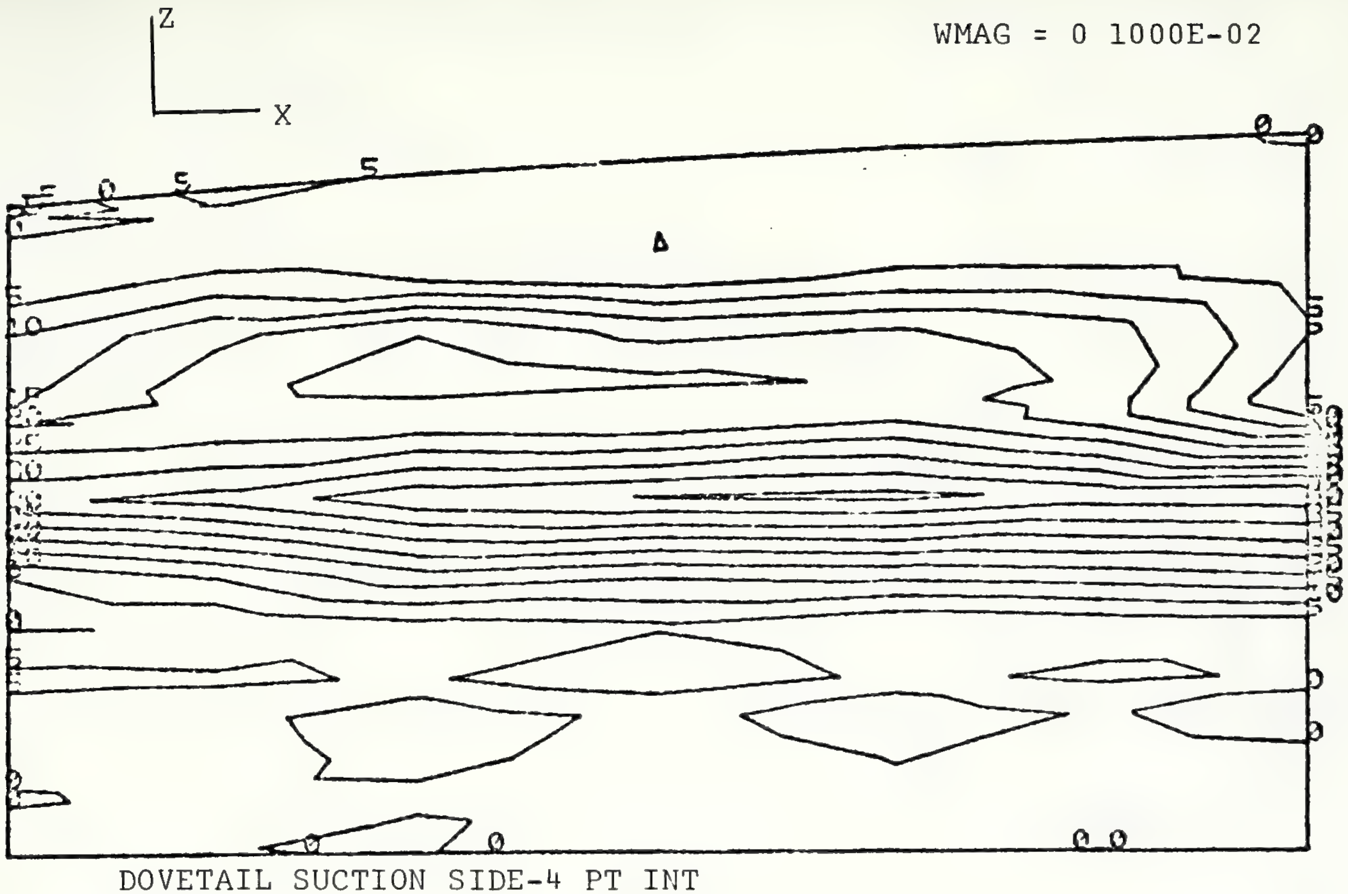
WMAG = 0 1000E-02



DOVETAIL SUCTION SIDE-2 PT INT

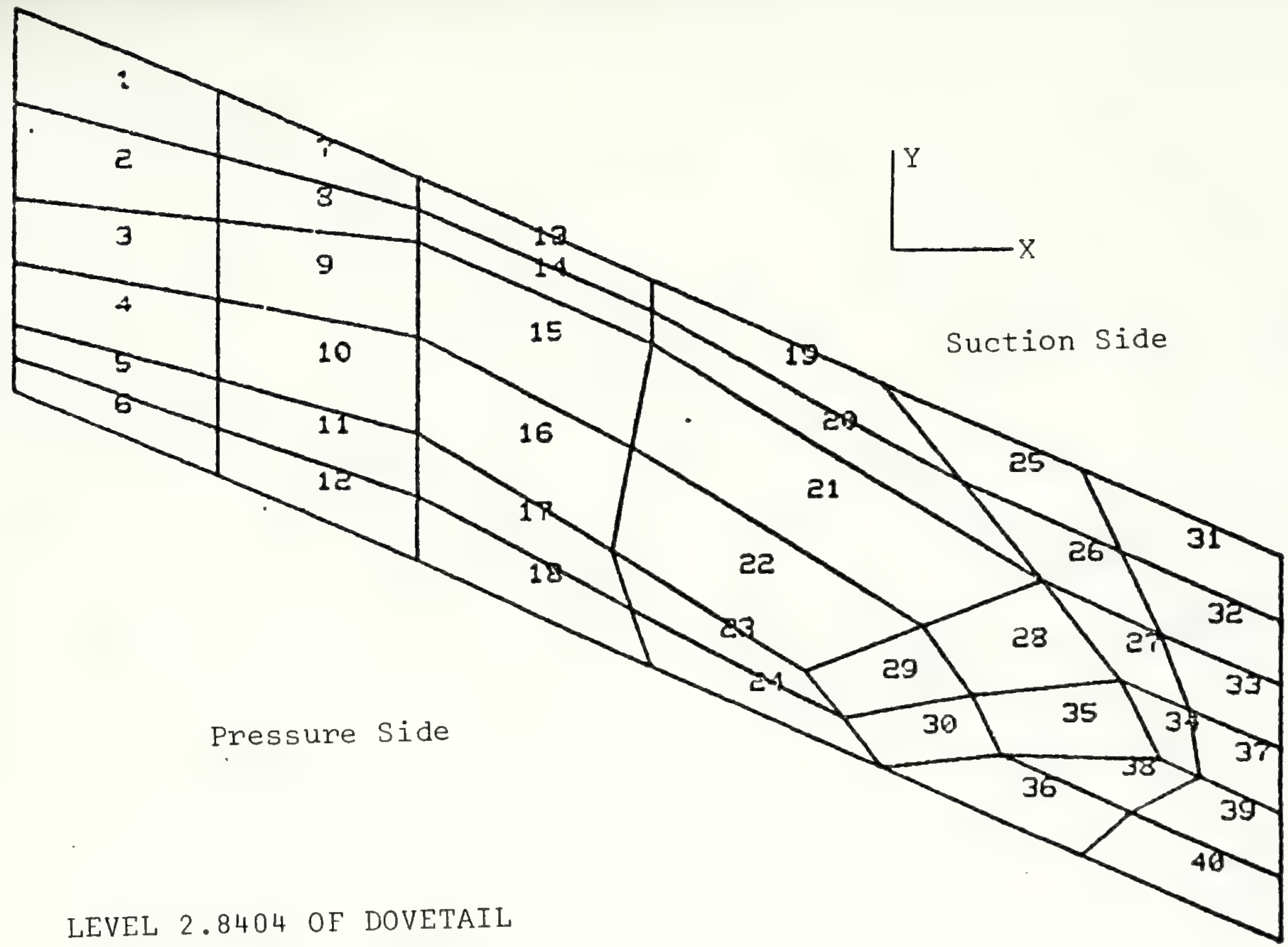
(b) Maximum Principal Stress Iso-lines for 2-pt Integration Order (ksi)

Figure 12. Continued, Page 2 of 3.



(c) Maximum Principal Stress Iso-lines for 4-pt Integration Order (ksi)

Figure 12. Continued, Page 3 of 3.

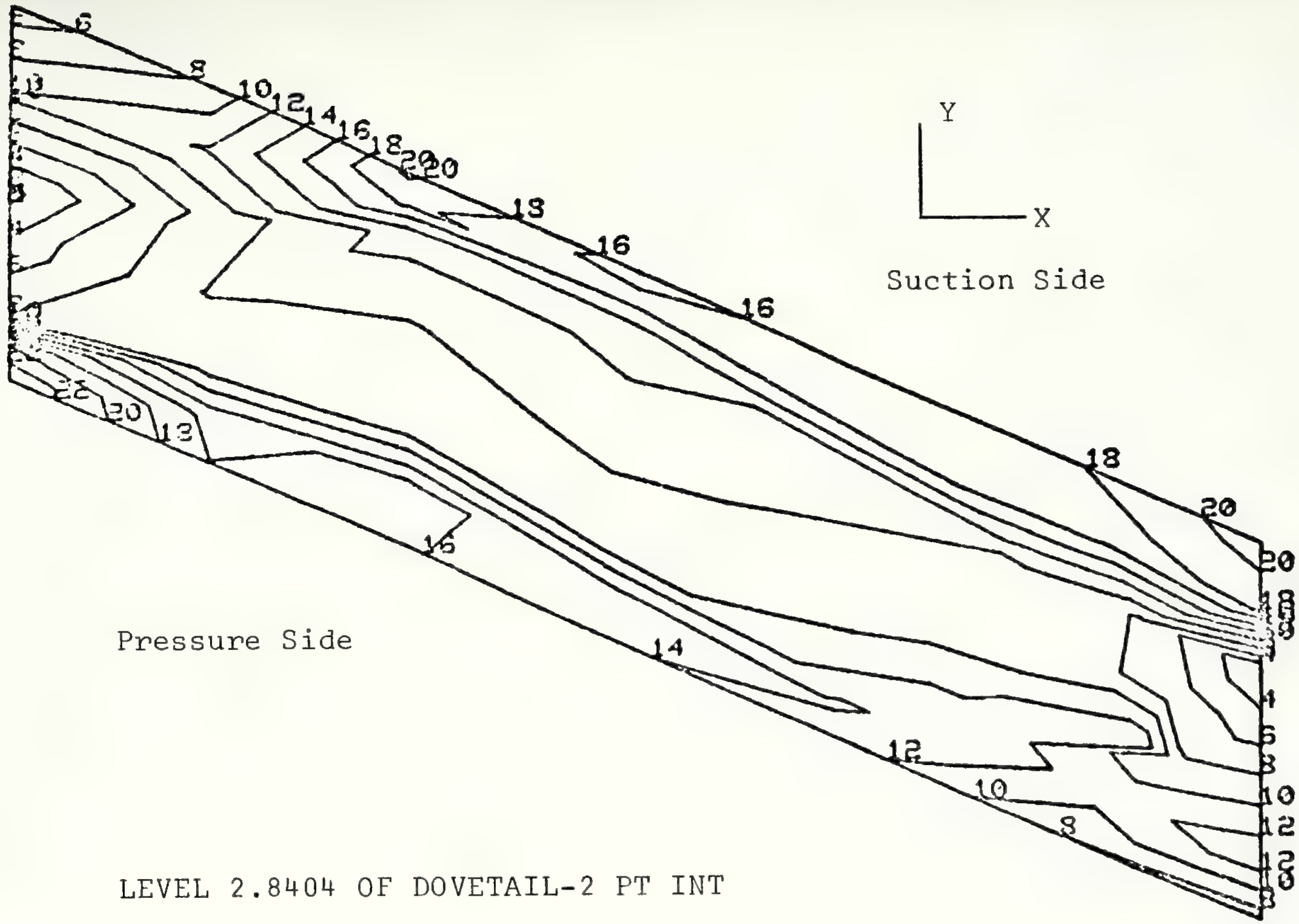


LEVEL 2.8404 OF DOVETAIL

(a) 2-D Element Arrangement

Figure 13. Level 2.8404, Page 1 of 3.

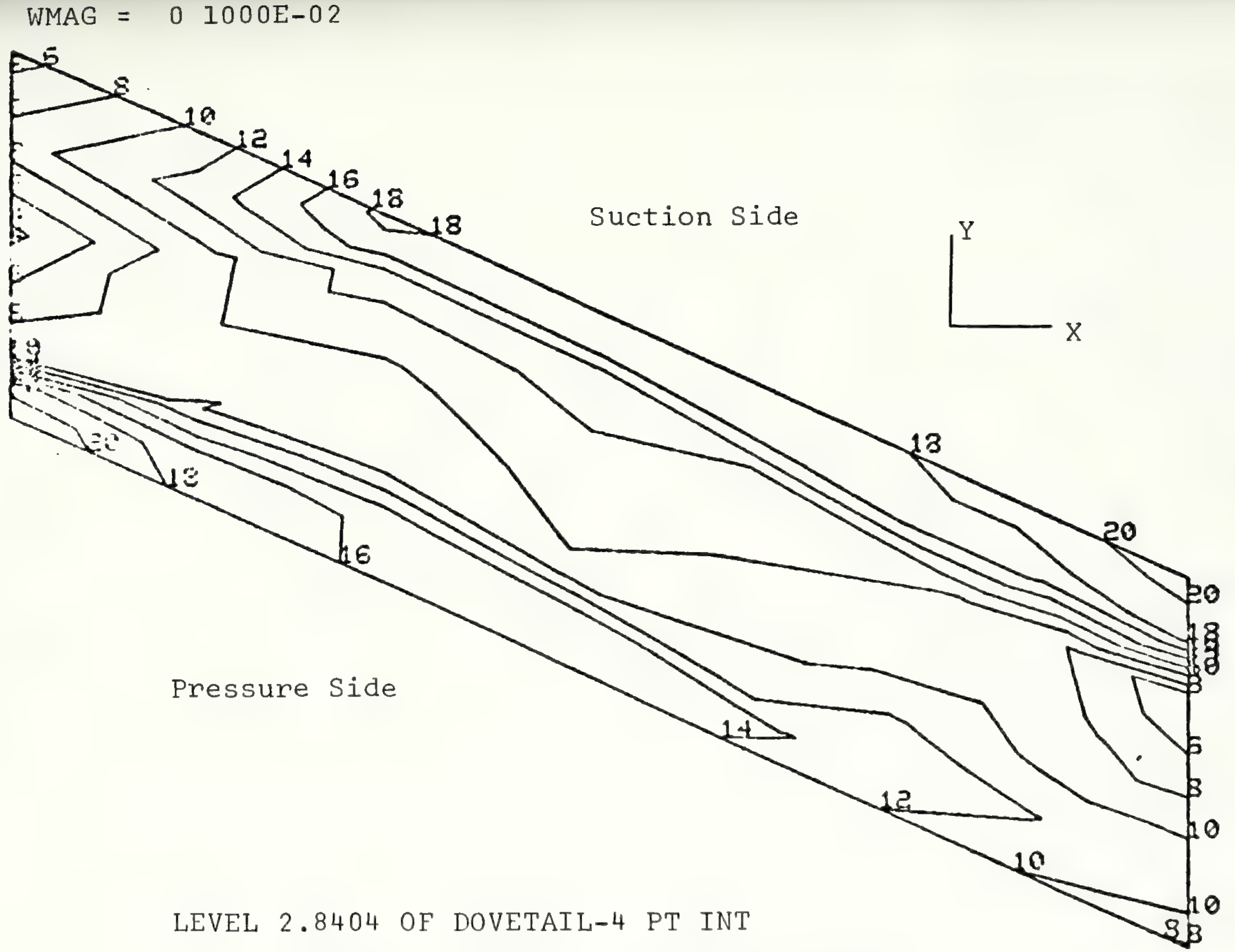
WMAG = 0 1000E-02



LEVEL 2.8404 OF DOVETAIL-2 PT INT

(b) Maximum Principal Stress Iso-lines
for 2-pt Integration Order (ksi)

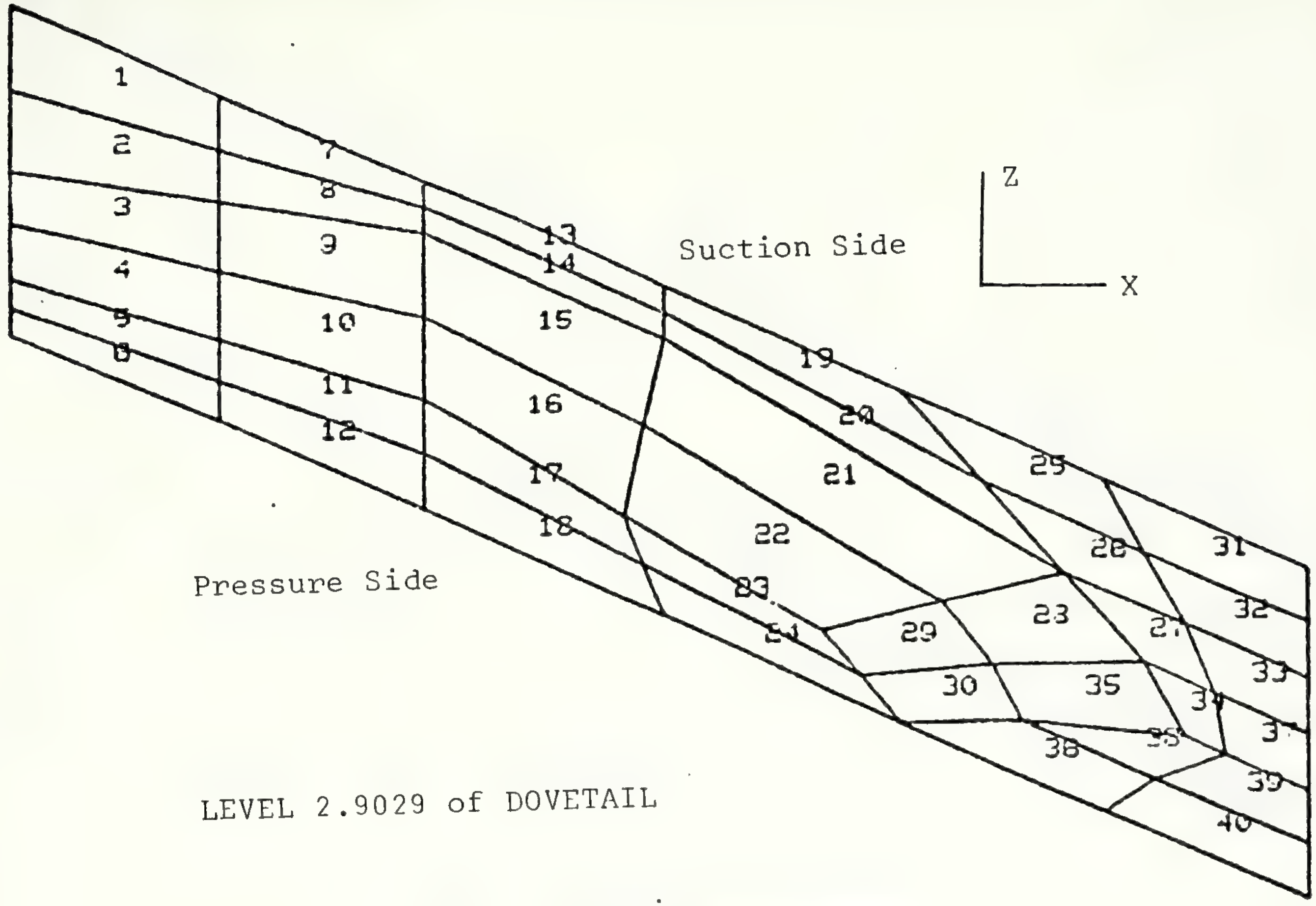
Figure 13. Continued, Page 2 of 3.



LEVEL 2.8404 OF DOVETAIL-4 PT INT

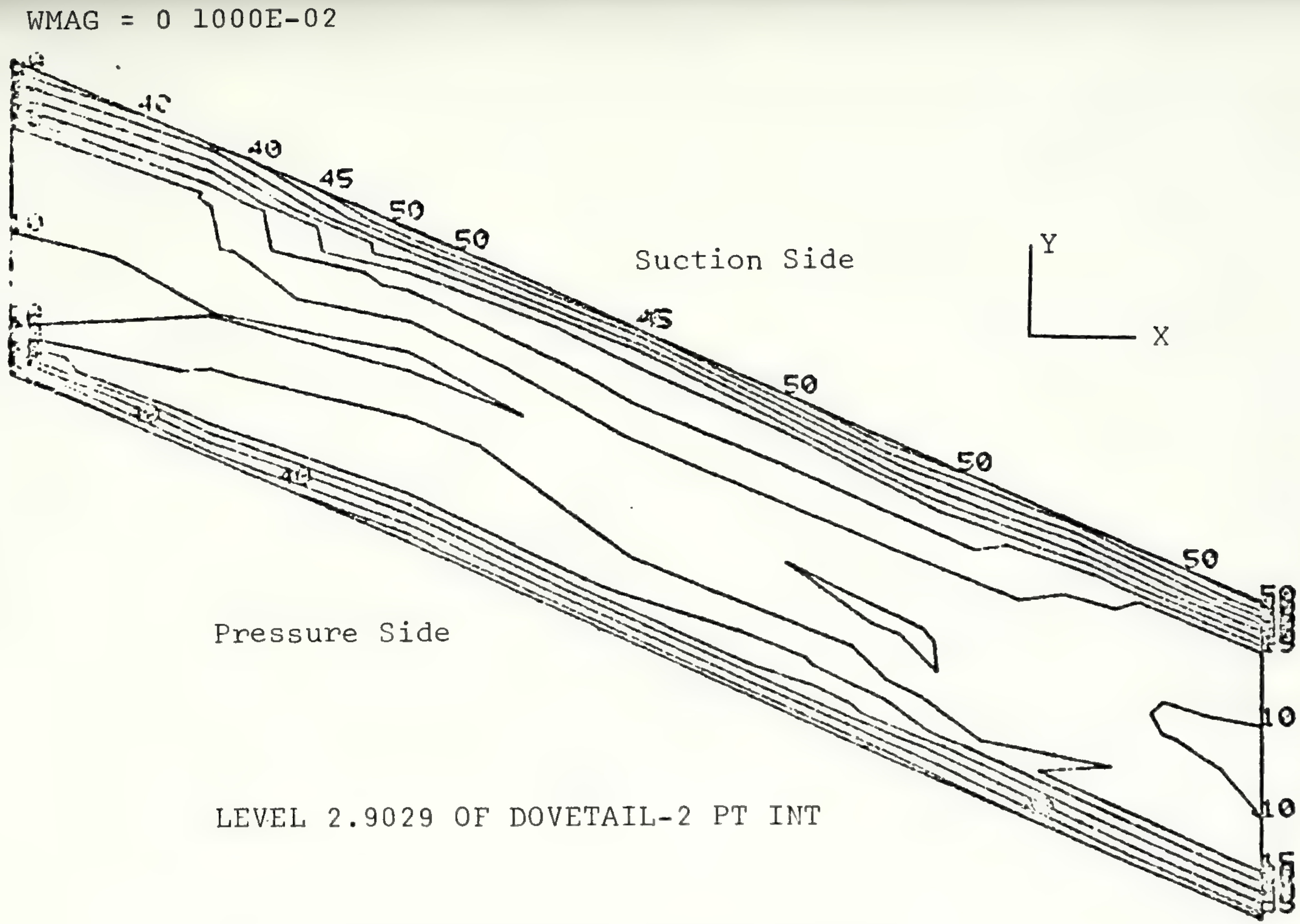
(c) Maximum Principal Stress Iso-lines for 4-pt Integration Order (ksi)

Figure 13. Continued, Page 3 of 3.



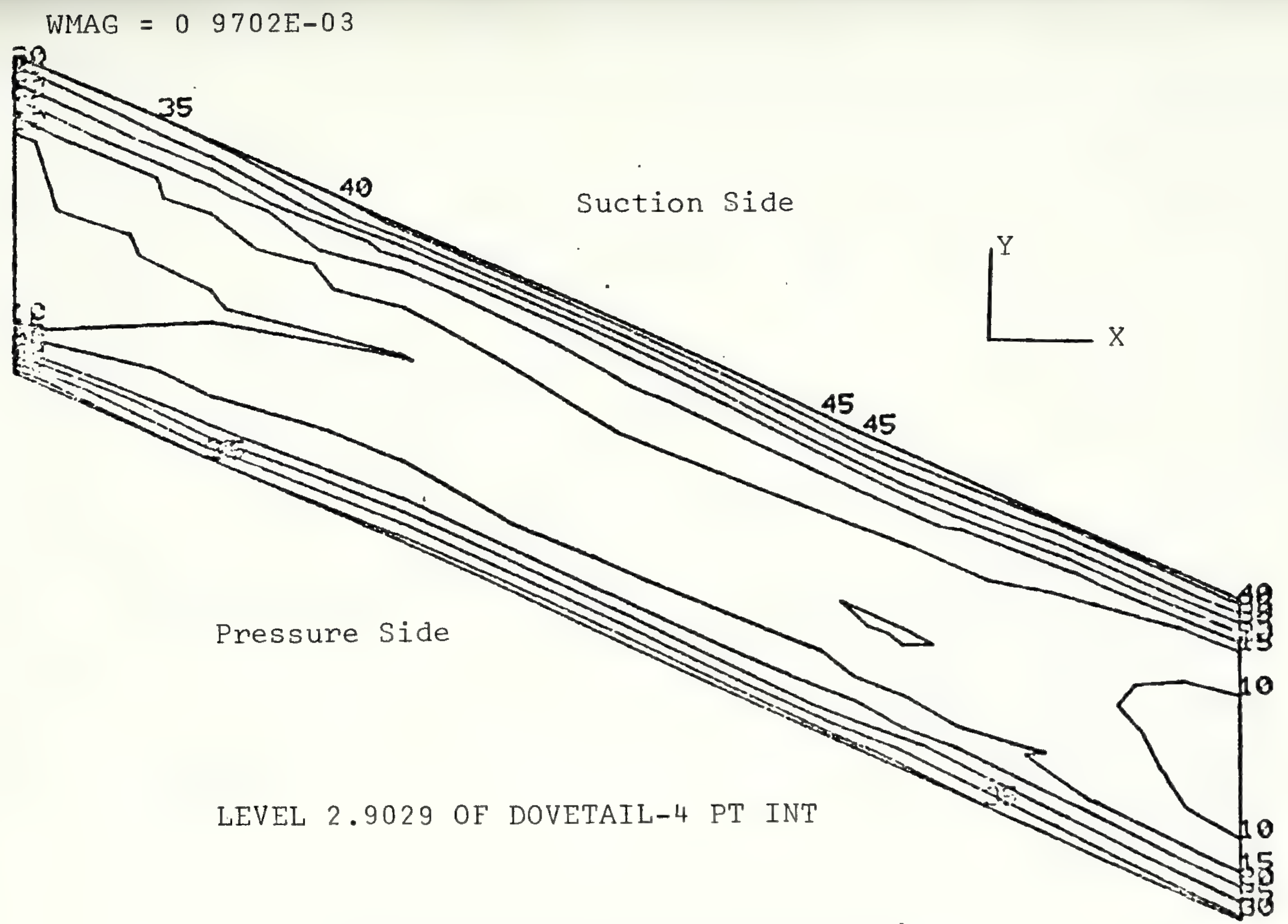
(a) 2-D Element Arrangement

Figure 14. Level 2.9029, Page 1 of 3.



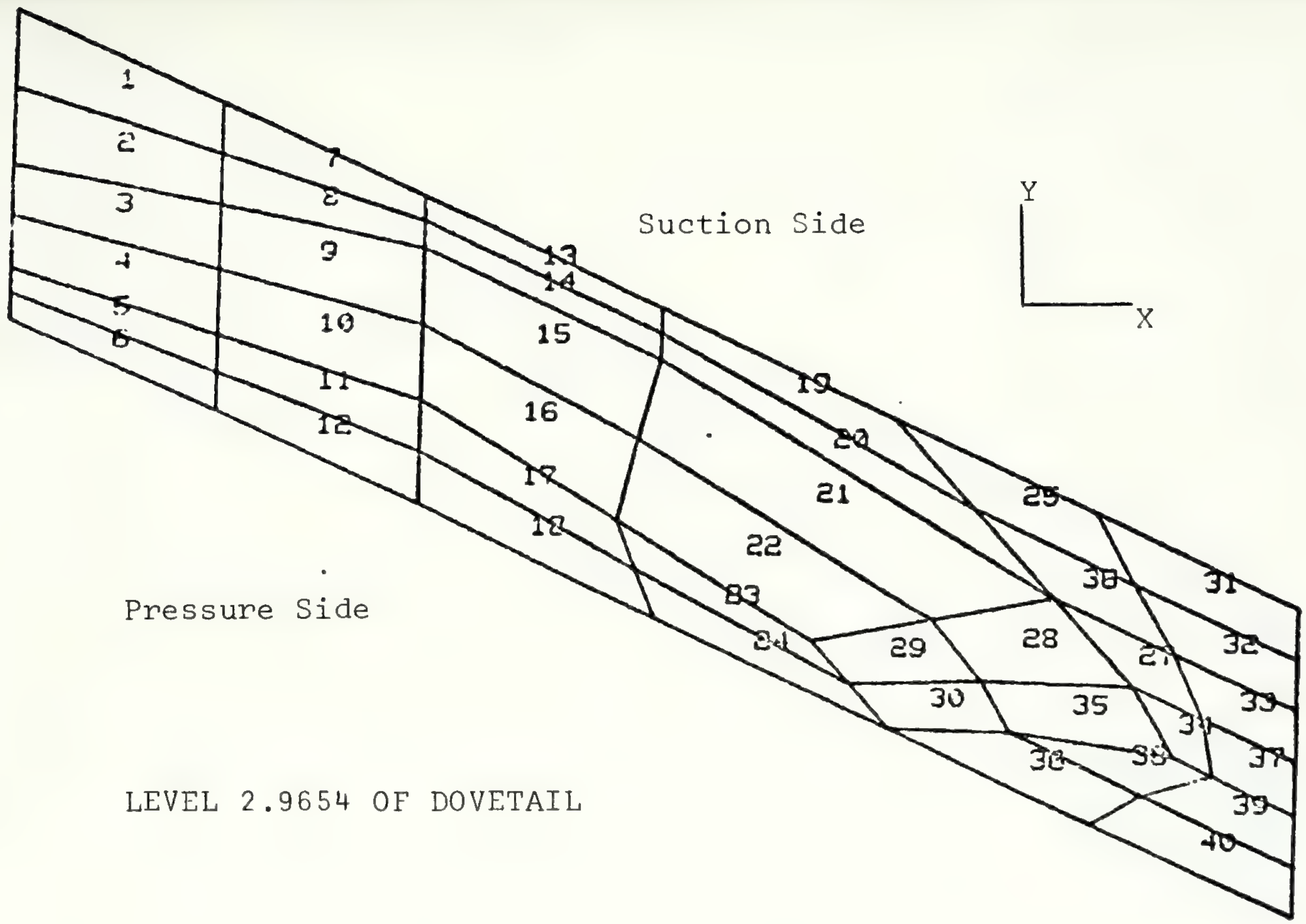
(b) Maximum Principal Stress Iso-lines for 2-pt Integration Order (ksi)

Figure 14. Continued, Page 2 of 3.



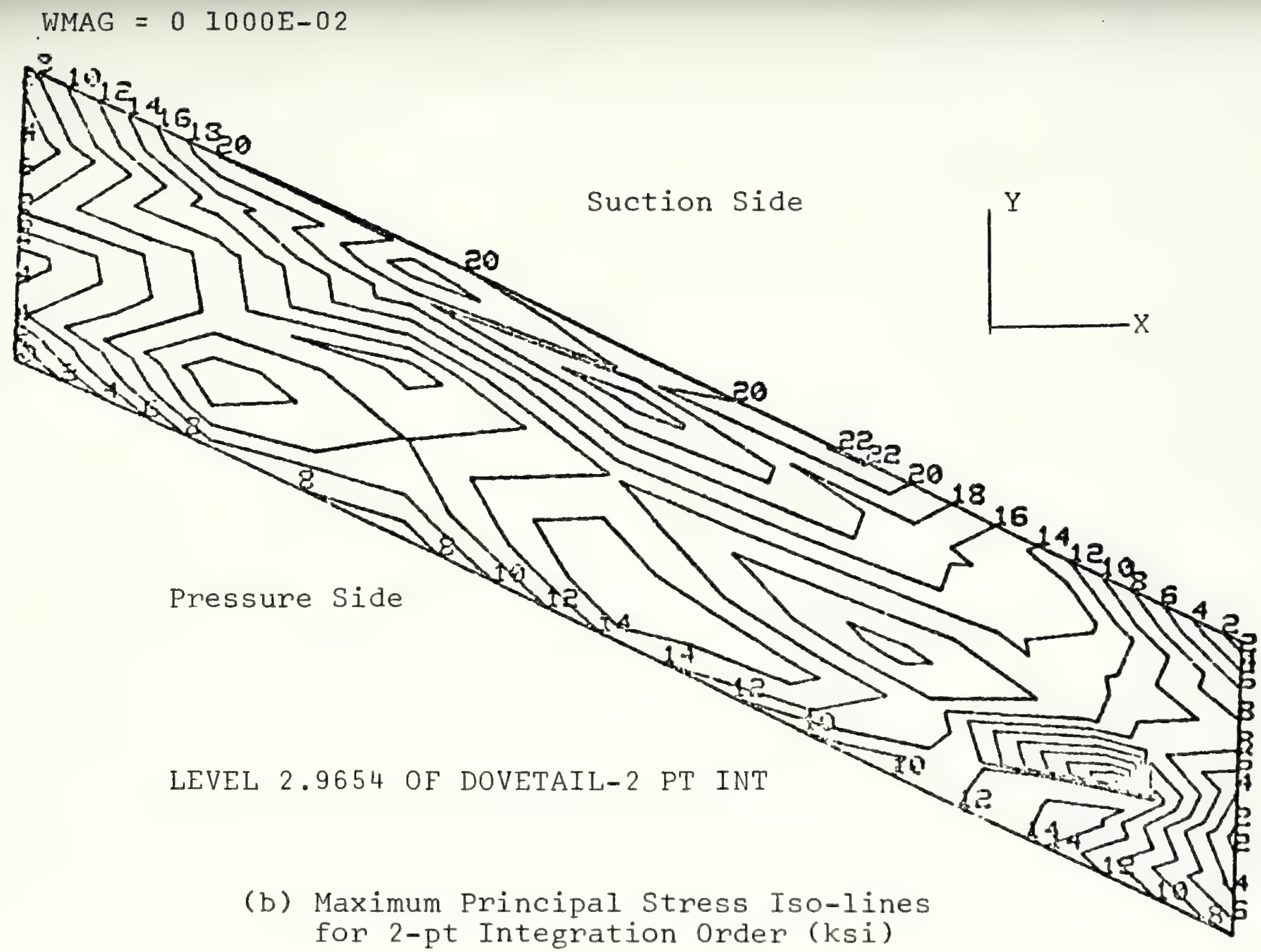
(c) Maximum Principal Stress Iso-lines
for 4-pt Integration Order (ksi)

Figure 14. Continued, Page 3 of 3.



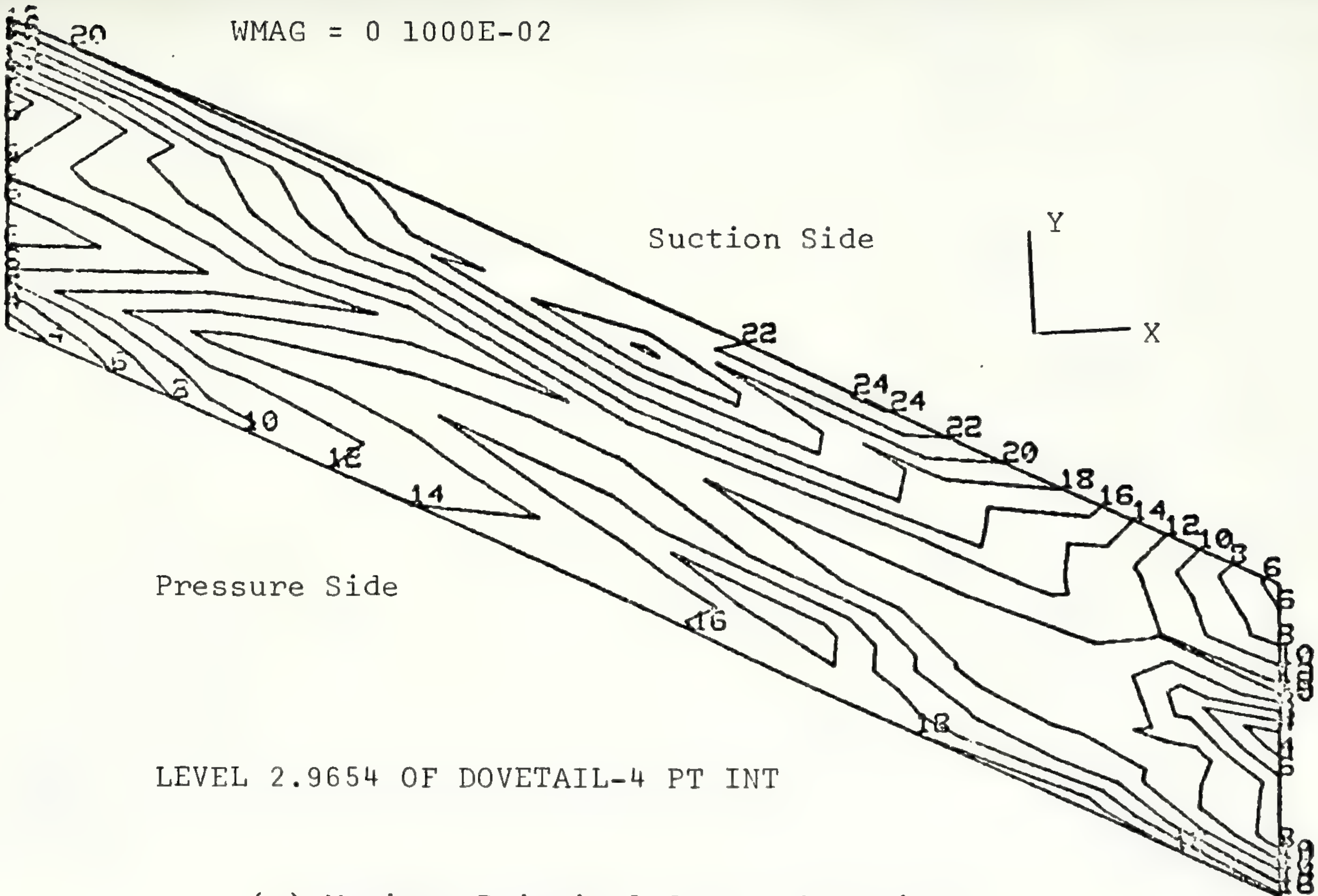
(a) 2-D Element Arrangement

Figure 15. Level 2.9651, Page 1 of 3.



(b) Maximum Principal Stress Iso-lines for 2-pt Integration Order (ksi)

Figure 15. Continued, Page 2 of 3.



(c) Maximum Principal Stress Iso-lines
for 4-pt Integration Order (ksi)

Figure 15. Continued, Page 3 of 3.

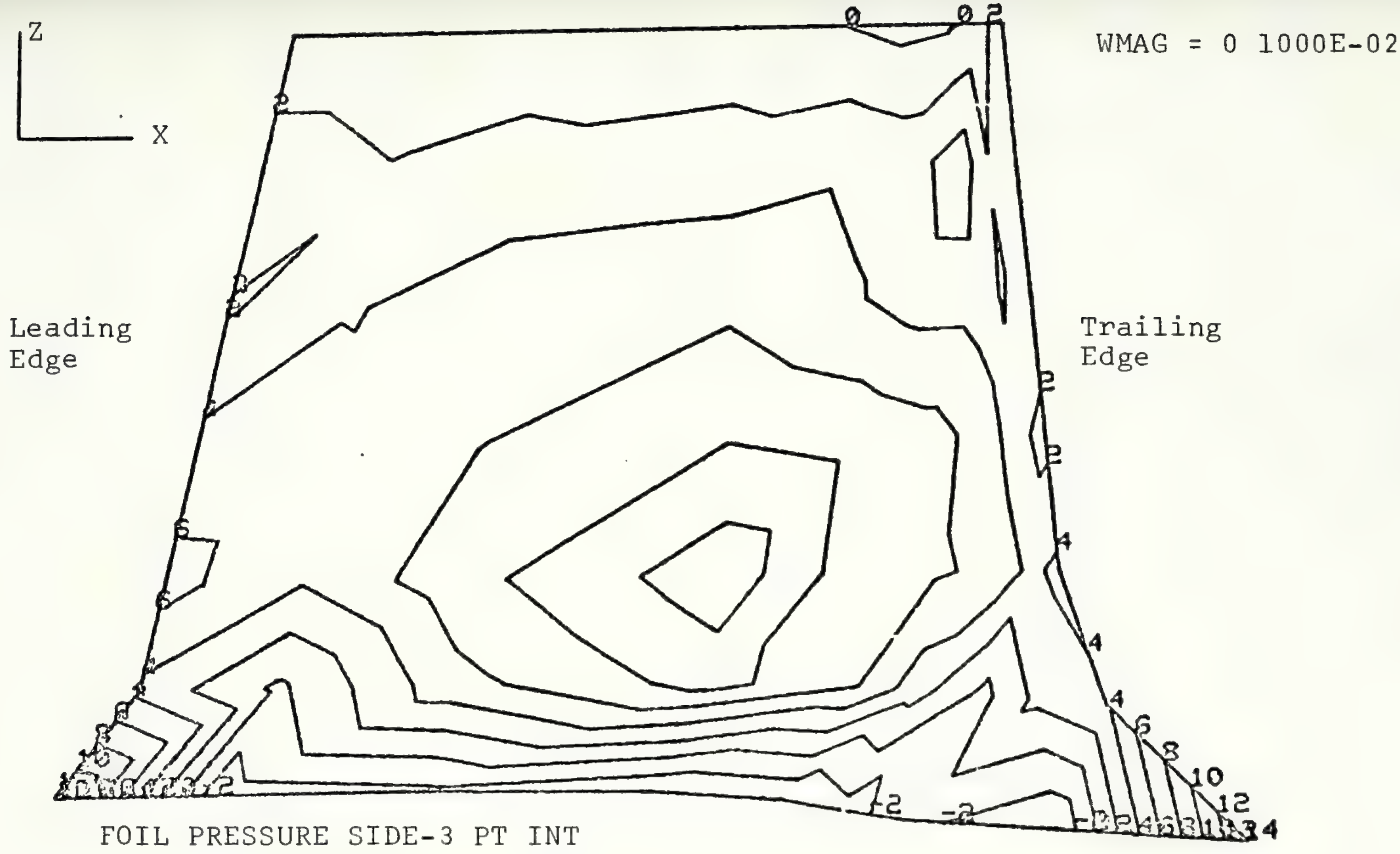


Figure 16. Maximum Principal Stress Contours for Pressure Side of Airfoil (ksi).

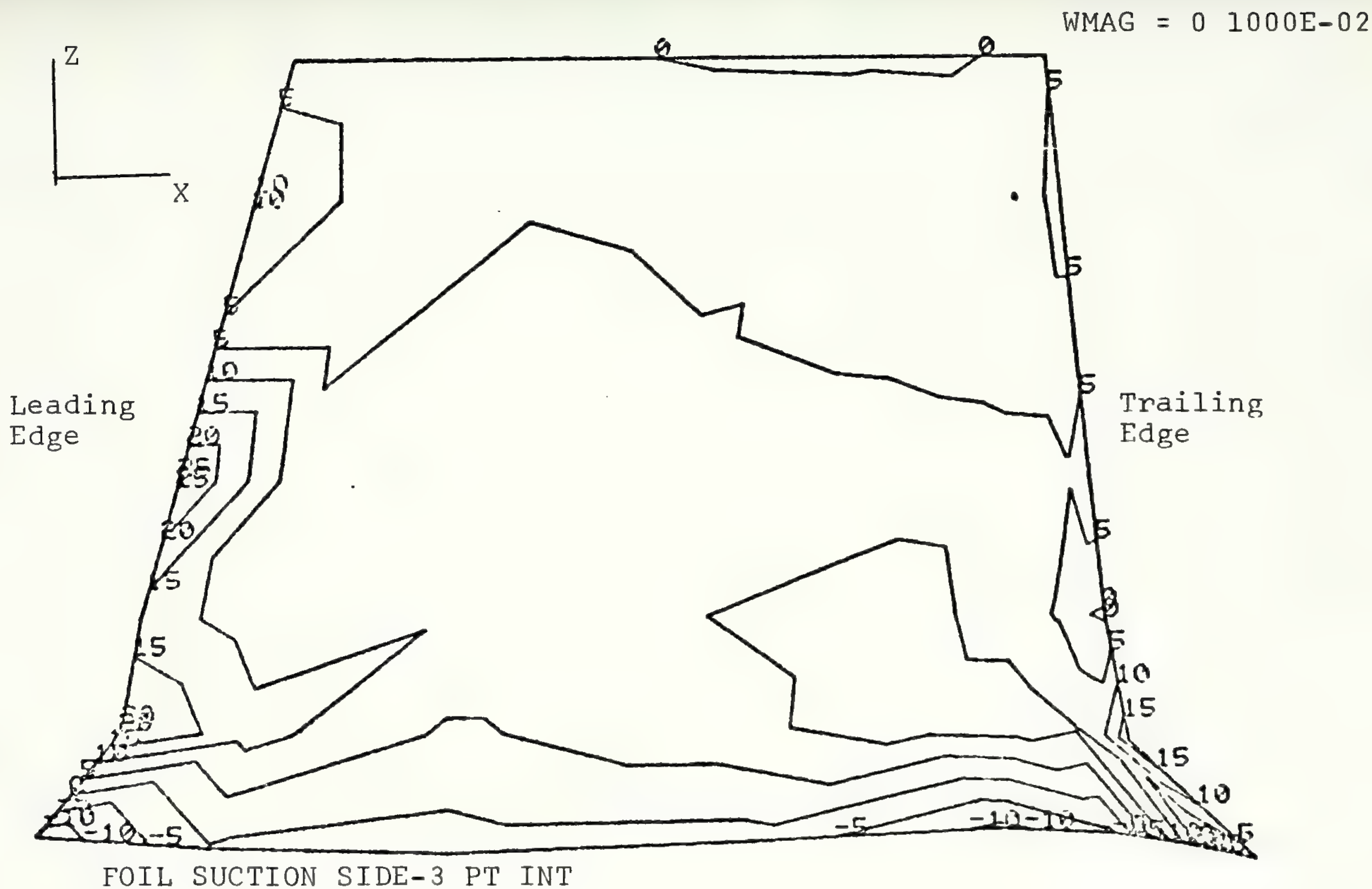


Figure 17. Maximum Principal Stress Contours for Suction Side of Airfoil (ksi).

VII. CONCLUSIONS

The time required for development of pre- and post-processors prevented a complete analysis which would have the potential for strong conclusive opinions and possible suggestions for design improvement. Software and hardware tools are, however, now implemented and identified which will allow follow-on analysis by NPS thesis students of geometric complex machine components. Despite the handicap of an inconclusive analysis, certain comments can be made concerning the results.

A. ANALYSIS RESULTS

Three other finite element analyses of ceramic turbine blade designs were reviewed by the author [Refs. 9, 10, 11]. Despite differing boundary conditions and loading schemes, a surface stress concentration persists in the region immediately above the blade-disk contact surface. This fact indicates that further research is required into the possibility of optimizing the dovetail geometry in order to spread the stresses over a greater bulk of material. If such an optimized design proves infeasible or non-existent, then the conclusion of this analysis and review indicates that hot-pressed Silicon-Nitride is unsuitable even for the limited goals of the current project. Using the four point bending strength of seventy-two ksi [Ref. 9] as a reference strength parameter, a simplistic factor of safety of 1.57 is realized. Considering the analysis did not impose

pressure and temperature gradient induced stresses, this factor of safety appears to be quite inadequate for predicting a decent probability of success.

Despite somewhat severe distortions in many of the analysis model elements, no mathematical singularities were encountered in using a reduced integration order. Further analysis using finer meshes may show that the reduced integration results are more accurate than the "exact" integration order. Should this be the case, approximately fifteen per cent higher values would be predicted than the results of "exact" integration of the stiffness matrix elements. These higher values would necessarily reduce the probability of success calculations normally used in design.

B. OPPORTUNITIES FOR FURTHER RESEARCH

The computer hardware and software at NPS, together with the groundwork laid by this thesis, allows for many avenues of follow-on research of which the following is a partial list:

1) refinement of the mesh of this analysis and convergence study of results,

2) analysis of the blade loaded under pressure and temperature gradient induced stresses,

3) optimization of attachment root design in order to alleviate stress concentrations,

4) probabilistic failure analysis using ADINA generated stresses,

5) frequency analysis of the blade,

6) investigation of appropriate boundary conditions,

7) analysis of the three material system of the blade-disk contact region.

APPENDIX A

ADDENDUM TO ADINA USER'S MANUAL, REPORT 82448-1, MIT, SEPTEMBER 1975 (Revised May 1976)

Two additional output options are available to the ADINA user at the Naval Postgraduate School besides what is described in the published user's manual. Master Control Card one, documented on page II.1, may be altered as follows:

NOTES	COLUMNS	VARIABLE	ENTRY
(a)	71-75	ITP57	Indicator for extended stress output and storage of stresses on user defined file 57. EQ.0; option not desired EQ.1; option desired
(b)	76-80	ITP58	Indicator for storage of displacements on user defined file 58. EQ.0; option not desired EQ.1; option desired

NOTES:

(a) The use of this option gives the user an output of stresses at all twenty-seven locations per element described in Section X.3 and illustrated on page X.37. Stresses are stored sequentially on user defined file 57 in the following order:

INPUT STEP

1. element number
2. integer 1 corresponding to figure X.5
3. normal stress for node in global X direction
4. normal stress for node in global Y direction
5. normal stress for node in global Z direction
6. shear stress XY
7. shear stress XZ
8. shear stress YZ

Steps 2 through 8 are repeated for the remainder of the twenty-seven nodes of the element, followed by a complete cycle for each of the remaining elements.

(b) Use of this option stores the six global displacements for each node on user defined file 58 in the following order:

INPUT STEP

1. node number
2. X direction displacement
3. Y direction displacement
4. Z direction displacement
5. X-axis rotation (if present)
6. Y-axis rotation (if present)
7. Z-axis rotation (if present)

Steps 1 through 7 are repeated for each of the input nodes. Displacements for the additional nodes for which stress results are calculated with option ITP57 are not calculated.

APPENDIX B

CONVERSION OF CALCOMP PLOTTING ROUTINES FOR USE ON A TEKTRONIX 4012 TERMINAL

The TEKTRONIX 4012 Computer Display Terminal is the most powerful interactive graphics system available to the general user at the Naval Postgraduate School. The major advantages of this unit are its speed and access to the IBM 360 computer via the CP/CMS system. A major disadvantage is there is only one terminal designated for the general user. During the final stages of preparation of this report, PSAP1 and Program Contour Plot were modified for use with the TEKTRONIX terminal in order to achieve a higher quality of plot in a timely manner than is possible with the CALCOMP system. These modifications proved quite simple and resulted in a usable plotting system. Presented in this appendix are the subroutines added to the graphics programs used by the author and a few general comments. The user should consult Refs. 12, 13, 14 and 15 for detailed information on using the TEKTRONIX unit.

The user must first determine the plotting origin of his routine and establish the limits of his plotting window using subroutine VWINDO. Caution must be used in order to ensure that the original program will properly scale the plotting values in both coordinate directions. CALCOMP PLOT statements which shift the origin must be deleted. Subroutines must then be added to the routine which are titled with CALCOMP plotting library routine names which in turn

call equivalent subroutines in the TEKTRONIX library. The following are examples used with PSAP1 and Program Contour Plot:

1) Subroutine equivalent to CALCOMP PLOTS:

```
      SUBROUTINE PLOTS
      IBAUD=480
      CALL INITT (IBAUD)
      CALL BINITT
      RETURN
      END
```

2) Subroutine equivalent to CALCOMP PLOTE:

```
      SUBROUTINE PLOTE
      IX=0
      IY=780
      CALL FINITT (IX,IY)
      RETURN
      END
```

3) Subroutine equivalent to CALCOMP LINE:

```
      SUBROUTINE LINE (Y,X,N,N1,N2)
      DIMENSION X(21), Y(21)
      CALL MOVEA (X(1),Y(1))
      DO 10 I=2,N
      CALL DRAWA (X(I),Y(I))
   10 CONTINUE
      RETURN
      END
```

4) Subroutine equivalent to CALCOMP NUMBER:

```
      SUBROUTINE NUMBER (Y,X,H,AL,TH,N)
      DIMENSION IARRAY (4)
      CALL MOVEA (X,Y)
      CALL IFORM (AL,4,IARRAY,32)
      DO 10 I=1,4
      IF (IARRAY(I).EQ.32) GO TO 10
      CALL ANCHO (IARRAY(I))
   10 CONTINUE
      RETURN
      END
```

5) Subroutine equivalent to CALCOMP SYMBOL:

```
      SUBROUTINE SYMBOL (IY,IX,H,BCK,TH,N)
      DIMENSION IARRAY (80),  BCK (1)
      CALL CONETA (BCK,N,IARRAY)
      CALL NOTATE (IX,IY,N,IARRAY)
      RETURN
      END
```

6) Subroutine equivalent to CALCOMP PLOT:

```
      SUBROUTINE PLOT (Y,X,I)
      IF (I.EQ.2) CALL DRAWA (X,Y)
      IF (I.EQ.3) CALL MOVEA (X,Y)
      RETURN
      END
```

These subroutines were tailored for use only with PSAP1 and Program Contour Plot and are much more restrictive than the CALCOMP equivalent. For example, subroutine NUMBER defined above will only plot integers up to four digits whereas subroutine NUMBER used with the CALCOMP will plot any integer or floating point number input by the user. The subroutines do, however, provide an example of the simple modifications required and the TEKTRONIX software is available for a much higher degree of graphics sophistication.

(This page intentionally blank)

(This page intentionally blank)

APPENDIX C

PROGRAM CENTRIFUGAL LOAD LISTING

```
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C      *
C      *  PROGRAM CENTRIFUGAL LOAD
C      *
C      *  THIS PROGRAM READS AN ADINA INPUT DECK AND CALCULATES NODAL
C      *  CONSISTANT LOADS CAUSED BY A GIVEN ROTATION ABOUT ANY COMBI-
C      *  NATION OF RIGHT-HAND CARTESIAN COORDINATE SYSTEM AXES.
C      *  VARIOUS CHOICES OF OUTPUTS CAN BE SELECTED INCLUDING A
C      *  PUNCHED CARD DECK IN A FORMAT ACCEPTABLE TO ADINA FINITE
C      *  ELEMENT CODE FOR INPUT OF CONCENTRATED NODAL FORCES.
C      *
C      *
C      *  INPUT DATA DECK
C      *
C      *  CARD 1  (FORMAT 4F10.0)
C      *
C      *     VARIABLE     COL      REMARKS
C      *      W1          1-10     ANGULAR VELOCITY ABOUT THE X-AXIS
C      *      W2         11-20     ANGULAR VELOCITY ABOUT THE Y-AXIS
C      *      W3         21-30     ANGULAR VELOCITY ABOUT THE Z-AXIS
C      *      RO         31-40     MASS DENSITY
C      *
C      *  CARD 2  (FORMAT 8I5)
C      *
C      *      NUMNF       1-5      NUMBER OF NODES
C      *      NUMEL       6-10     NUMBER OF ELEMENTS
C      *      NGP        11-15     ORDER OF QAUSS POINT INTEGRATION (2-6)
C      *      NCUR       16-20     LOAD CURVE NUMBER TO BE USED IN ADINA
C      *                           IAW SECTION XII OF ADINA USERS MANUAL
C      *      ICHK1      21 25     FLAG FOR PRINTOUT OF NODE COORDINATE AND
C      *                           MESH CONECTIVITY DATA
C      *                             0-NO PRINTOUT
C      *                             1-PRINTOUT
C      *      ICHK2      26-30     FLAG FOR PRINTOUT OF CONSISTANT LOADS
C      *                           BY ELEMENT
C      *                             0-NO PRINTOUT
C      *                             1-PRINTOUT
C      *      ICHK3      31-35     FLAG FOR PRINTOUT OF TOTALED CONSISTANT
C      *                           LOADS FOR EACH NODE
C      *                             0-NO PRINTOUT
```

```
C      *                          1-PRINTOUT                             CEN00410
C      *      ICHK4     36-40    FLAG FOR PUNCHED CECK CF NODE CONSISTANTCEN00420
C      *                         FCRCES                                  CEN00430
C      *                           0-NO PLNCHING                         CEN00440
C      *                           1-PUNCH LOAD DECK                     CEN00450
C      *                                                                 CEN00460
C      *   PLACE ADINA INPUT DECK HERE.  THE PORTION OF THE DECK REQUIRED CEN00470
C      *   IS FROM THE TITLE CARD TO THE LAST CONNECTIVITY CARD INCLUSIVE.CEN00480
C      *                                                                 CEN00490
C      *   A DYNAMIC DIMENSIONING SCHEME IS USED IN ORDER TO GENERALIZE THCEN00500
C      *   THE PROGRAM FOR VARIOUS MESHES.  SHOULC MCRE STORAGE SPACE    CEN00510
C      *   BE REQUIRED, THE FOLLOWING DIAGNOSTIC WOLLD BE CUTPUT:        CEN00520
C      *         MTCT=XXXXXX    INCREASE MTOT FOR ACDITICNAL STCRAGE     CEN00530
C      *   SHOULD THIS DIAGNOSTIC OCCUR, CHANGE THE FCLLCWING TWC CARCS  CEN00540
C      *   IN THE MAIN PROGRAM:                                          CEN00550
C      *       COMMCN A(XXXXXX)                                          CEN00560
C      *       MTOT=XXXXXXX                                              CEN00570
C      *   INCREASE XXXXXX TO A VALUE GREATER THAN THAT GIVEN FOR MTOT   CEN00580
C      *   BY THE DIAGNOSTIC.                                            CEN00590
C      *                                                                 CEN00600
C      *   REFERENCES:                                                   CEN00610
C      *     BATHE,KLAUS-JURGEN; ADINA(REPORT 82448-1);SEPT 1975,        CEN00620
C      *       (REVISION MAY 1976);MIT, CAMBRIDGE MASS.                  CEN00630
C      *                                                                 CEN00640
C      *     EASTERLING,L.R.;FINITE ELEMENT ANALYSIS CF A CERAMIC GAS    CEN00650
C      *       TURBINE BLADE, THESIS REPORT; NPS,MONTEREY CA. MAR 1978   CEN00660
C      *                                                                 CEN00670
C      *   WRITTEN BY:  PROF GILLES CANTIN AND L.R. EASTERLING,          CEN00680
C      *        NPS, MONTEREY CA. FEBRUARY 1978.                         CEN00690
C      *                                                                 CEN00700
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CEN00710
C                                                                        CEN00720
       IMPLICIT REAL*8 (A-H,O-Z)                                         CEN00730
C                                                                        CEN00740
C      ****************************************************************  CEN00750
C                                                                        CEN00760
C         MAIN PROGRAM TO READ ADINA DECK AND CALCLLATE CCNSISTANT       CEN00770
C         CENTRIFUCAL LOADS                                              CEN00780
C                                                                        CEN00790
C      ****************************************************************  CEN00800
C                                                                        CEN00820
       CCMMON A(15C00)                                                   CEN00830
       CCMMON/RPARA/W1,W2,W3,RO                                          CEN00840
       COMMON/IPARA/NUMNP,NUMEL,NGP,NCUR                                 CEN00850
       CCMMON/CHECK/ICHK1,ICHK2,ICHK3,ICHK4                              CEN00860
       MTOT=15000                                                        CEN00870
       READ(5,1000)W1,W2,W3,RO                                           CEN00880
 1000  FCRMAT(4F10.0)                                                    CEN00890
```

```
      READ(5,1001)NUMNP,NUMEL,NGP,NCUR,ICHK1,ICHK2,ICHK3,ICHK4
 1001 FORMAT(8I5)
      WRITE(6,1002)W1,W2,W3,RO
      WRITE(6,1003)NUMNP,NUMEL,NGP
 1002 FORMAT(//5X,'ANGULAR VELOCITIES'/10X,'WX = ',G25.16/10X,'WY = ',
     1 G25.16/10X,'WZ = ',G25.16//5X,'DENSITY = ',G25.16/)
 1003 FORMAT(//5X,'NUMBER OF NODE POINTS = ',I5//5X,'NUMBER OF
     2 ELEMENT = ',I5//5X,'NUMBER OF GAUSS POINTS FOR INTERGRATION
     3 = ',I5/)
      N1=1
      N2=N1+60*NUMEL
      N3=N2+NUMNP
      N4=N3+NUMNP
      N5=N4+NUMNP
      N6=N5+NUMNP
      N7=N6+NUMEL
      N8=N7+20*NUMEL
      NLAST=N8+3*NUMNP
      IF(NLAST.GT.MTOT)WRITE(6,2000)NLAST
 2000 FORMAT(//5X, 'MTOT = ',I5,'   INCREASE MTOT FOR ADDITIONAL STORAGE
     1'//)
      CALL CENTF(A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8))
      STOP
      END
      SUBROUTINE CENTF(TFVT,XPT,YPT,ZPT,NUMPT,NREL,NCONT,FORCE)
C
C     ***************************************************************
C
C        SUBROUTINE TO COORDINATE INPUT AND OUTPUT OF
C        CONSISTANT CENTRIFUGAL LOADS.
C
C     ***************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/ RPARA/W1,W2,W3,RO
      COMMON/ IPARA/NUMNP,NUMEL,NGP,NCUR
      COMMON/ LOADS/NLOAD,NLCUR,NPTM,IDGRAV
      COMMON/CHECK/ICHK1,ICHK2,ICHK3,ICHK4
      DIMENSION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1)
      DIMENSION TFVT(20,3,1),FORCE(3,1),NCONT(20,1)
      CALL RDADIN(XPT,YPT,ZPT,NCONT,NREL,NUMPT)
   60 IF(ICHK1.EQ.0)GO TO 10
      CALL PNTOUT(XPT,YPT,ZPT,NCONT,NREL,NUMPT)
   10 CALL TRANS(TFVT,XPT,YPT,ZPT,NCONT,NREL,NUMPT,FORCE)
      RETURN
      END
      SUBROUTINE TRANS (TFVT,XPT,YPT,ZPT,NCONT,NREL,NUMPT,FORCE)
C
```

```
C
C     ****************************************************************
C
C        SUBROUTINE TO TRANSFER PARAMETERS FROM CENTF TO CUBAT
C
C     ****************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/RPARA/W1,W2,W3,RO
      COMMON/CHECK/ICHK1,ICHK2,ICHK3,ICHK4
      COMMON/IPARA/NUMNP,NUMEL,NGP,NCUR
      COMMON/LOADS/NLOAD,NLCUR,NPTM,IDGRAV
      DIMENSION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1),NCONT(20,1)
     1 ,TFVT(20,3,1),FORCE(3,1)
      DIMENSION FVT(20,3),NCON(20),COD(20,3)
      DO 10 I=1,NUMEL
      DO 20 J=1,20
      NCON(J)=NCONT(J,I)
      K=NCON(J)
      COD(J,1)=XPT(K)
      COD(J,2)=YPT(K)
      COD(J,3)=ZPT(K)
   20 CONTINUE
      CALL CUBAT(COD,FVT,W1,W2,W3,RO,NCON,NGP,NREL(I))
      DO 30 N=1,3
      DO 40 II=1,20
      IF(DABS(FVT(II,N)).LT.0.1D-8) GO TO 50
      TFVT(II,N,I)=FVT(II,N)
      GO TO 40
   50 TFVT(II,N,I)=0.0D0
   40 CONTINUE
   30 CONTINUE
      IF(ICHK2.EQ.0)GO TO 10
      WRITE(6,1000)I
      WRITE(6,1001)
      DO 60 J=1,20
      WRITE(6,1002)J,NCON(J),(FVT(J,N),N=1,3)
   60 CONTINUE
   10 CONTINUE
  222 FORMAT(//5X,'XTOT = ',G25.16//5X,'YTOT = ',G25.16//5X,'ZTOT = ',
     1 G25.16/)
      DO 444 I=1,NUMNP
      DO 444 J=1,3
  444 FORCE(J,I)=0.0D0
      DO 333 I=1,NUMEL
      DO 333 J=1,20
      L=NCONT(J,I)
      DO 333 K=1,3
  333 FORCE(K,L)=FORCE(K,L)+TFVT(J,K,I)
```

```
      XTOT2=0.0D0
      YTOT2=0.0D0
      ZTOT2=0.0D0
      DO 555 I=1,NUMNP
      XTOT2=XTOT2+FORCE(1,I)
      YTOT2=YTOT2+FORCE(2,I)
  555 ZTOT2=ZTOT2+FORCE(3,I)
      IF(ICHK3.EQ.0)GO TO 888
      DO 666 I=1,NUMNP
      WRITE(6,1006)I,(FORCE(J,I),J=1,3)
 1006 FORMAT(//4X,'NODE ',I5,2X,'FX = ',G25.16,2X,'FY = ',G25.16,2X,
     1 'FZ = ',G25.16)
  666 CONTINUE
  888 WRITE(6,222)XTOT2,YTOT2,ZTOT2
      KLOAD=0
      IF(ICHK4.EQ.0)GO TO 999
      DO 777 I=1,NUMNP
      DO 777 J=1,3
      IF(DABS(FORCE(J,I)).LT.0.1D-8)GO TO 777
      KLOAD=KLOAD+1
      WRITE(7,1004)I,J,NCUR,FORCE(J,I)
  777 CONTINUE
      WRITE(6,1003)KLOAD
 1000 FORMAT(//5X,'CONSISTANT CENTRIFUGAL LOADS FOR ELEMENT  ',I5//)
 1001 FORMAT(5X,'NATURAL',5X,'GLOBAL'/5X,'COORD NR',4X,'COORD NR ',7X,'
     1 X-COMPONENT',14X,'Y-COMPONENT',14X,'Z-COMPONENT'//)
 1002 FORMAT('0',6X,I5,6X,I5,3G30.16)
 1003 FORMAT(//5X,'TOTAL ADDITIONAL NLOAD REQUIRED IS  ',I10/)
 1004 FORMAT(3I5,G25.16)
 1005 FORMAT(4I5)
      RETURN
      END
      SUBROUTINE CUBAT(COD,FVT,W1,W2,W3,RO,NCON,NGP,IEL)
C
C         CUBAT FORMS THE CONSISTENT LOAD VECTOR CORRESPONDING TO THE
C         ANGULAR VELOCITY OF COMPONENT W1,W2,W3 ABOUT THE ORIGIN
C         PROF GILLES CANTIN AT THE NAVAL POSTGRDXLATE SCHOOL SEPT. 1977
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION COD(20,3),FVT(20,3),DUM(20,3),GP(6,5),WF(6,5),NCON(20)
      DATA GP/.577350269189630D0,-.577350269189630D0,0.0D0,0.0D0,0.0D0,0.D
     10,.774596669241480D0,0.0D0,-.774596669241480D0,0.0D0,0.0D0,0.0D0,.86
     21136311594050D0,0.339981043584860D0,-.339981043584860D0,-.86113631159
     3405D0,0.0D0,0.0D0,.906179845938660D0,.538469310105680D0,0.0D0,-.5384
     469310105680D0,-.906179845938660D0, 0.0D0,.932469514203150D0,.66120938
     5646626D0,.238619186083170D0,-.238619186083170D0,-.661209386466260D0,-
     6.932469514203150D0/
      DATA WF/1.0D0,1.0D0,0.0D0,0.0D0,0.0D0,0.0D0,.555555555555560D0,.888
```

```
     1888888888889D0,.5555555555555556D0,0.0D0,0.0D0,0.0D0,.34785484513745D
     20,.65214515486255D0,.65214515486255D0,.34785484513745D0,0.0D0,0.0D
     30,.23692688505619D0,.47862867049937D0,.56888888888889D0,.47862867O
     449937D0,.23692688505619D0,0.0D0,.17132449237917D0,.36076157304814D
     50,.46791393457269D0,.46791393457269D0,.36076157304814D0,.171324492
     637917D0/
      IF(NGP.LE.0.OR.NGP.GT.6) NGP=3
      ZRO =0.0D0
      DO 100 I=1,20
      DO 100 J=1,3
  100 FVT(I,J) = ZRO
      IDGP = NGP-1
      DO 200 I=1,NGP
      R = GP(I,IDGP)
      WR= WF(I,IDGP)
      DO 200 J=1,NGP
      S = GP(J,IDGP)
      WS= WF(J,IDGP)
      DO 200 K=1,NGP
      T = GP(K,IDGP)
      WT= WF(K,IDGP)
      WFAC = WR*WS*WT
      CALL SHP20(R,S,T,DTJ,DUM,COD,W1,W2,W3,NCON,IEL)
      DO 150 L =1,20
      DO 150 M=1,3
  150 FVT(L,M) = FVT(L,M) + DUM(L,M)*WFAC
  200 CONTINUE
      DO 300 I=1,20
      DO 300 J=1,3
  300 FVT(I,J)=FVT(I,J)*RO
      RETURN
      END
      SUBROUTINE SHP20(R,S,T,DTJ,FVT,COD,W1,W2,W3,NCON,IEL)
C
C         UPON EXIT FROM THE PRESENT SUBROUTINE, FVT WILL CONTAIN
C         THE VALUES OF LOAD VECTOR ARGUMENT FVT(20,3) THAT CAN BE
C         USED FOR GAUSSIAN QUADRATURE
C
C         THE MATRIX ARGUMENT IS THE RESULT OF THE PRODUCT OF THE
C         FOLLOWING MATRICES:  NT(20,3)*A(3,3)*N(3,20)*XYZ(20,3)*DETJ
C         WHERE THE MATRIX N(3,20) IS THE MATRIX OF SHAPE FUNCTIONS
C         NT(20,3) IS THE TRANSPOSE OF N(3,20), XYZ(20,3) IS THE
C         MATRIX OF COORDINATES OF THE NODES AND DETJ IS THE DETERMINANT
C         OF THE JACOBIAN MATRIX
C
C         R,S,T ARE THE NON DIMENSIONAL COORDINATES OF THE ELEMENT
C         DTJ IS THE DETERMINANT OF THE JACOBIAN MATRIX
C         FVT  IS THE OUTPUT VECTOR
```

```
C         COD  IS AN INPUT ARRAY OF THE ELEMENT COCRCINATES IN THE        CEN02820
C         GLOBAL CCORDINAT SYSTEM                                         CEN02830
C         W1 IS THE COMPONENT AF ANGULAR VELOCITY ABOUT THE AXIS X        CEN02840
C         W2 IS THE COMPCNENT AF ANGULAR VELOCITY ABCUT THE AXIS Y        CEN02850
C         W3 IS THE COMPCNENT AF ANGULAR VELOCITY ABOUT THE AXIS Z        CEN02860
C         NCON IS THE CONNECTIVITY MATRIX OF THE ELEMENT                  CEN02870
C         IEL  IS THE ELEMENT IDENTIFICATION NUMBER                       CEN02880
C                                                                         CEN02890
C         PROF. GILLES CANTIN  SEPT. 1977  NAVAL PCSTGRADUATE SCHOOL      CEN02900
C         MONTEREY CALIFCRNIA  93940                                      CEN02910
C                                                                         CEN02920
      IMPLICIT REAL*8(A-H,O-Z)                                            CEN02930
      DIMENSION COD(20,3),DNI(3,20),       EMI(20),PRCD1(3),PROD2(3)     CEN02940
      DIMENSION FVT(20,3),A(3,3),NCON(20)                                 CEN02950
      REAL*8 JAC(3,3)                                                     CEN02960
      GP(BT) = 0.5D0*(1.0D0 + BT)                                         CEN02970
      GM(BT) = 0.5D0*(1.0D0 - BT)                                         CEN02980
      GZ(BT) = 1.0D0 - BT*BT                                              CEN02990
      DP(BT) =  0.5D0+ZRC*BT                                              CEN03000
      DM(BT) = -0.5D0+ZRC*BT                                              CEN03010
      DZ(BT) = -2.0D0*BT                                                  CEN03020
      ZRO = 0.0D0                                                         CEN03030
      ONE = 1.0D0                                                         CEN03040
      TWO = 2.0D0                                                         CEN03050
      DO 100 I=1,3                                                        CEN03060
      DO 100 J=1,20                                                       CEN03070
      FVT(J,I) = ZRO                                                      CEN03080
  100 DNI(I,J) = ZRO                                                      CEN03090
      EMI( 9) = GZ(R)*GP(S)*GP(T)                                         CEN03100
      IF(NCON( 9).EQ.0) EMI( 9) = ZRO                                     CEN03110
      EMI(10) = GM(R)*GZ(S)*GP(T)                                         CEN03120
      IF(NCON(10).EQ.0) EMI(10) = ZRO                                     CEN03130
      EMI(11) = GZ(R)*GM(S)*GP(T)                                         CEN03140
      IF(NCON(11).EQ.0) EMI(11) = ZRO                                     CEN03150
      EMI(12) = GP(R)*GZ(S)*GP(T)                                         CEN03160
      IF(NCON(12).EQ.0) EMI(12) = ZRO                                     CEN03170
      EMI(13) = GZ(R)*GP(S)*GM(T)                                         CEN03180
      IF(NCON(13).EQ.0) EMI(13) = ZRO                                     CEN03190
      EMI(14) = GM(R)*GZ(S)*GM(T)                                         CEN03200
      IF(NCON(14).EQ.0) EMI(14) = ZRO                                     CEN03210
      EMI(15) = GZ(R)*GM(S)*GM(T)                                         CEN03220
      IF(NCON(15).EQ.0) EMI(15) = ZRO                                     CEN03230
      EMI(16) = GP(R)*GZ(S)*GM(T)                                         CEN03240
      IF(NCON(16).EQ.0) EMI(16) = ZRO                                     CEN03250
      EMI(17) = GP(R)*GP(S)*GZ(T)                                         CEN03260
      IF(NCON(17).EQ.0) EMI(17) = ZRO                                     CEN03270
      EMI(18) = GM(R)*GP(S)*GZ(T)                                         CEN03280
      IF(NCON(18).EQ.0) EMI(18) = ZRO                                     CEN03290
```

```
      EMI(19) = GM(R)*GM(S)*GZ(T)                                      CEN03300
      IF(NCON(19).EQ.0) EMI(19) = ZRO                                  CEN03310
      EMI(20) = GP(R)*GM(S)*GZ(T)                                      CEN03320
      IF(NCON(20).EQ.0) EMI(20) = ZRO                                  CEN03330
      EMI( 1) = GP(R)*GP(S)*GP(T) - (EMI( 9) + EMI(12) + EMI(17))/TWO  CEN03340
      EMI( 2) = GM(R)*GP(S)*GP(T) - (EMI( 9) + EMI(10) + EMI(18))/TWO  CEN03350
      EMI( 3) = GM(R)*GM(S)*GP(T) - (EMI(10) + EMI(11) + EMI(19))/TWO  CEN03360
      EMI( 4) = GP(R)*GM(S)*GP(T) - (EMI(11) + EMI(12) + EMI(20))/TWO  CEN03370
      EMI( 5) = GP(R)*GP(S)*GM(T) - (EMI(13) + EMI(16) + EMI(17))/TWO  CEN03380
      EMI( 6) = GM(R)*GP(S)*GM(T) - (EMI(13) + EMI(14) + EMI(18))/TWO  CEN03390
      EMI( 7) = GM(R)*GM(S)*GM(T) - (EMI(14) + EMI(15) + EMI(19))/TWO  CEN03400
      EMI( 8) = GP(R)*GM(S)*GM(T) - (EMI(15) + EMI(16) + EMI(20))/TWO  CEN03410
      IF(NCON( 9).EQ.0) GO TO 500                                      CEN03420
      DNI(1, 9) = DZ(R)*GP(S)*GP(T)                                    CEN03430
      DNI(2, 9) = GZ(R)*DP(S)*GP(T)                                    CEN03440
      DNI(3, 9) = GZ(R)*GP(S)*DP(T)                                    CEN03450
  500 IF(NCON(10).EQ.0) GO TO 510                                      CEN03460
      DNI(1,10) = DM(R)*GZ(S)*GP(T)                                    CEN03470
      DNI(2,10) = GM(R)*DZ(S)*GP(T)                                    CEN03480
      DNI(3,10) = GM(R)*GZ(S)*DP(T)                                    CEN03490
  510 IF(NCON(11).EQ.0) GO TO 520                                      CEN03500
      DNI(1,11) = DZ(R)*GM(S)*GP(T)                                    CEN03510
      DNI(2,11) = GZ(R)*DM(S)*GP(T)                                    CEN03520
      DNI(3,11) = GZ(R)*GM(S)*DP(T)                                    CEN03530
  520 IF(NCON(12).EQ.0) GO TO 530                                      CEN03540
      DNI(1,12) = DP(R)*GZ(S)*GP(T)                                    CEN03550
      DNI(2,12) = GP(R)*DZ(S)*GP(T)                                    CEN03560
      DNI(3,12) = GP(R)*GZ(S)*DP(T)                                    CEN03570
  530 IF(NCON(13).EQ.0) GO TO 540                                      CEN03580
      DNI(1,13) = DZ(R)*GP(S)*GM(T)                                    CEN03590
      DNI(2,13) = GZ(R)*DP(S)*GM(T)                                    CEN03600
      DNI(3,13) = GZ(R)*GP(S)*DM(T)                                    CEN03610
  540 IF(NCON(14).EQ.0) GO TO 550                                      CEN03620
      DNI(1,14) = DM(R)*GZ(S)*GM(T)                                    CEN03630
      DNI(2,14) = GM(R)*DZ(S)*GM(T)                                    CEN03640
      DNI(3,14) = GM(R)*GZ(S)*DM(T)                                    CEN03650
  550 IF(NCON(15).EQ.0) GO TO 560                                      CEN03660
      DNI(1,15) = DZ(R)*GM(S)*GM(T)                                    CEN03670
      DNI(2,15) = GZ(R)*DM(S)*GM(T)                                    CEN03680
      DNI(3,15) = GZ(R)*GM(S)*DM(T)                                    CEN03690
  560 IF(NCON(16).EQ.0) GO TO 570                                      CEN03700
      DNI(1,16) = DP(R)*GZ(S)*GM(T)                                    CEN03710
      DNI(2,16) = GP(R)*DZ(S)*GM(T)                                    CEN03720
      DNI(3,16) = GP(R)*GZ(S)*DM(T)                                    CEN03730
  570 IF(NCON(17).EQ.0) GO TO 580                                      CEN03740
      DNI(1,17) = DP(R)*GP(S)*GZ(T)                                    CEN03750
      DNI(2,17) = GP(R)*DP(S)*GZ(T)                                    CEN03760
      DNI(3,17) = GP(R)*GP(S)*DZ(T)                                    CEN03770
```

```
580 IF(NCON(18).EQ.0) GO TO 590                                          CEN03780
    DNI(1,18) = DM(R)*GP(S)*GZ(T)                                        CEN03790
    DNI(2,18) = GM(R)*DP(S)*GZ(T)                                        CEN03800
    DNI(3,18) = GM(R)*GP(S)*DZ(T)                                        CEN03810
590 IF(NCON(19).EQ.0) GO TO 600                                          CEN03820
    DNI(1,19) = DM(R)*GM(S)*GZ(T)                                        CEN03830
    DNI(2,19) = GM(R)*DM(S)*GZ(T)                                        CEN03840
    DNI(3,19) = GM(R)*GM(S)*DZ(T)                                        CEN03850
600 IF(NCON(20).EQ.0) GO TO 620                                          CEN03860
    DNI(1,20) = DP(R)*GM(S)*GZ(T)                                        CEN03870
    DNI(2,20) = GP(R)*DM(S)*GZ(T)                                        CEN03880
    DNI(3,20) = GP(R)*GM(S)*DZ(T)                                        CEN03890
620 CONTINUE                                                             CEN03900
    DNI(1,1) = DP(R)*GP(S)*GP(T) - (DNI(1, 9)+DNI(1,12)+DNI(1,17))/TWOCEN03910
    DNI(2,1) = GP(R)*DP(S)*GP(T) - (DNI(2, 9)+DNI(2,12)+DNI(2,17))/TWOCEN03920
    DNI(3,1) = GP(R)*GP(S)*DP(T) - (DNI(3, 9)+DNI(3,12)+DNI(3,17))/TWOCEN03930
    DNI(1,2) = DM(R)*GP(S)*GP(T) - (DNI(1, 9)+DNI(1,10)+DNI(1,18))/TWOCEN03940
    DNI(2,2) = GM(R)*DP(S)*GP(T) - (DNI(2, 9)+DNI(2,10)+DNI(2,18))/TWOCEN03950
    DNI(3,2) = GM(R)*GP(S)*DP(T) - (DNI(3, 9)+DNI(3,10)+DNI(3,18))/TWOCEN03960
    DNI(1,3) = DM(R)*GM(S)*GP(T) - (DNI(1,10)+DNI(1,11)+DNI(1,19))/TWOCEN03970
    DNI(2,3) = GM(R)*DM(S)*GP(T) - (DNI(2,10)+DNI(2,11)+DNI(2,19))/TWOCEN03980
    DNI(3,3) = GM(R)*GM(S)*DP(T) - (DNI(3,10)+DNI(3,11)+DNI(3,19))/TWOCEN03990
    DNI(1,4) = DP(R)*GM(S)*GP(T) - (DNI(1,11)+DNI(1,12)+DNI(1,20))/TWOCEN04000
    DNI(2,4) = GP(R)*DM(S)*GP(T) - (DNI(2,11)+DNI(2,12)+DNI(2,20))/TWOCEN04010
    DNI(3,4) = GP(R)*GM(S)*DP(T) - (DNI(3,11)+DNI(3,12)+DNI(3,20))/TWOCEN04020
    DNI(1,5) = DP(R)*GP(S)*GM(T) - (DNI(1,13)+DNI(1,16)+DNI(1,17))/TWOCEN04030
    DNI(2,5) = GP(R)*DP(S)*GM(T) - (DNI(2,13)+DNI(2,16)+DNI(2,17))/TWOCEN04040
    DNI(3,5) = GP(R)*GP(S)*DM(T) - (DNI(3,13)+DNI(3,16)+DNI(3,17))/TWOCEN04050
    DNI(1,6) = DM(R)*GP(S)*GM(T) - (DNI(1,13)+DNI(1,14)+DNI(1,18))/TWOCEN04060
    DNI(2,6) = GM(R)*DP(S)*GM(T) - (DNI(2,13)+DNI(2,14)+DNI(2,18))/TWOCEN04070
    DNI(3,6) = GM(R)*GP(S)*DM(T) - (DNI(3,13)+DNI(3,14)+DNI(3,18))/TWOCEN04080
    DNI(1,7) = DM(R)*GM(S)*GM(T) - (DNI(1,14)+DNI(1,15)+DNI(1,19))/TWOCEN04090
    DNI(2,7) = GM(R)*DM(S)*GM(T) - (DNI(2,14)+DNI(2,15)+DNI(2,19))/TWOCEN04100
    DNI(3,7) = GM(R)*GM(S)*DM(T) - (DNI(3,14)+DNI(3,15)+DNI(3,19))/TWOCEN04110
    DNI(1,8) = DP(R)*GM(S)*GM(T) - (DNI(1,15)+DNI(1,16)+DNI(1,20))/TWOCEN04120
    DNI(2,8) = GP(R)*DM(S)*GM(T) - (DNI(2,15)+DNI(2,16)+DNI(2,20))/TWOCEN04130
    DNI(3,8) = GP(R)*GM(S)*DM(T) - (DNI(3,15)+DNI(3,16)+DNI(3,20))/TWOCEN04140
    DO 630 I=1,3                                                         CEN04150
    DO 630 J=1,3                                                         CEN04160
    JAC(I,J)=ZRO                                                         CEN04170
    DO 630 K=1,20                                                        CEN04180
630 JAC(I,J) = JAC(I,J) + DNI(I,K)*CCC(K,J)                              CEN04190
    A1 = JAC(1,1)*JAC(2,2)*JAC(3,3)                                      CEN04200
    A2 = JAC(1,2)*JAC(2,3)*JAC(3,1)                                      CEN04210
    A3 = JAC(1,3)*JAC(2,1)*JAC(3,2)                                      CEN04220
    B1 = JAC(3,1)*JAC(2,2)*JAC(1,3)                                      CEN04230
    B2 = JAC(3,2)*JAC(2,3)*JAC(1,1)                                      CEN04240
    B3 = JAC(3,3)*JAC(2,1)*JAC(1,2)                                      CEN04250
```

```
      DTJ = A1+A2+A3 - (B1+B2+B3)
      IF(DTJ.LE.ZRO) GO TO 960
      A(1,1) = W2*W2 + W3*W3
      A(2,2) = W3*W3 + W1*W1
      A(3,3) = W1*W1 + W2*W2
      A(1,2) = -W1*W2
      A(1,3) = -W1*W3
      A(2,3) = -W2*W3
      A(2,1) = A(1,2)
      A(3,1) = A(1,3)
      A(3,2) = A(2,3)
      DC 700 I=1,3
      PROD1(I)=0.CD0
      DC 700 J=1,20
  70C PROD1(I)=PROD1(I)+EMI(J)*COD(J,I)
      DC 800 I=1,3
      PRCD2(I)=0.0D0
      DC 800 J=1,3
  800 PRCD2(I)=PROD2(I)+A(I,J)*PROD1(J)
      DC 900 I=1,20
      DC 900 J=1,3
  900 FVT(I,J)=EMI(I)*PROD2(J)
      DC 950 I=1,20
      DO 950 J=1,3
  950 FVT(I,J) = FVT(I,J)*DTJ
      RETURN
  960 WRITE(6,1000)IEL,DTJ,R,S,T
 1000 FORMAT(//,5X,'ELEMENT NUMBER ',I5,' HAS A JACOBIAN DETERMINANT',/
     1,5X,'OF VALUE ',1G25.16,'*** EXECUTION STOPS ***'/,
     2 5X,' R = ',G25.16/5X,' S = ',G25.16/5X,' T = ',G25.16//)
      STOP
      END
      SUBROUTINE PNTOUT(XPT,YPT,ZPT,NCCNT,NREL,NUMPT)
C
C     ***********************************************************
C
C        SUBROUTINE TO PRINT CUT NODE ANC ELEMENT INFORMATICN
C
C     ***********************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      CCMMON/IPARA/NUMNP,NUMEL,NGP,NCUR
      DIMENSION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1),NCCNT(20,1)
C
C *** FCR OUTPUT CF GEOMETRY INFORMATICN
C
      WRITE(6,16)
   16 FORMAT(///,5X,'GRID PCINT INFORMATION',///)
```

```
      WRITE(6,17)                                                        CEN04740
   17 FCRMAT(5X,'RESEQUENCED',4X,'USER INPUT'/                           CEN04750
     15X,'GRID POINT',5X,'GRID POINT'/                                   CEN04760
     25X,'NUMBER',9X,'NUMBER',13X,'X',14X,'Y',14X,'Z'//)                 CEN04770
      DO 30 I=1,NUMNP                                                    CEN04780
      WRITE(6,18) I,NUMPT(I),XPT(I),YPT(I),ZPT(I)                        CEN04790
   18 FCRMAT(2X,I10,5X,I10,3X,3E15.4)                                    CEN04800
   30 CCNTINUE                                                           CEN04810
      WRITE(6,19)                                                        CEN04820
   19 FCRMAT(///,5X,'ELEMENT INFORMATICN - WITH RESEQUENCED GRID PCINTS CEN04830
     1'///)                                                              CEN04840
      WRITE (6,9008)                                                     CEN04850
 9008 FCRMAT(1X,'RESEQUENCED',4X,'USER INPUT',25X,'GRID POINTS'/         CEN04860
     11X,'ELEMENT',8X,'ELEMENT'/                                         CEN04870
     21X,'NUMBER',9X,'NUMBER',7X,'    1    2    3    4    5    6    7    CEN04880
     3 8    9   10   11   12   13   14   15   16   17   18   19   20'//)CEN04890
      DC 40 I=1,NUMEL                                                    CEN04900
      WRITE(6,9010)I,NREL(I),(NCONT(J,I),J=1,20)                         CEN04910
 9010 FORMAT(1X,I4,11X,I4,9X,20I5)                                       CEN04920
   40 CCNTINUE                                                           CEN04930
  999 RETURN                                                             CEN04940
      END                                                                CEN04950
      SUBROUTINE RDADIN(XPT,YPT,ZPT,NCCNT,NREL,NUMPT)                    CEN04960
C                                                                        CEN04970
C     ****************************************************************   CEN04980
C                                                                        CEN04990
C        SUBROUTINE TO READ ADINA DECK.                                  CEN05000
C                                                                        CEN05010
C     ****************************************************************   CEN05020
C                                                                        CEN05030
      IMPLICIT REAL*8 (A-H,O-Z)                                          CEN05040
      REAL*4 TITLE(20)                                                   CEN05050
      CCMMON/IPARA/NUMNP,NUMEL,NGP,NCUR                                  CEN05060
      CCMMON/LOADS/NLOAC,NLCUR,NPTM,IDGRAV                               CEN05070
      DIMENSICN XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1),NCONT(20,1)        CEN05080
      DIMENSICN IDOF(6),ID(6),IDOLD(6),NODE(20),NFAR(20)                 CEN05090
     1 ,NP(20),INP(20)                                                   CEN05100
      DATA CTEST/'X       '/                                             CEN05110
      NCARD=0                                                            CEN05120
      READ(5,9000)(TITLE(I),I=1,20)                                      CEN05130
      WRITE(6,9011)(TITLE(I),I=1,20)                                     CEN05140
 9011 FCRMAT(///5X,20A4,///)                                             CEN05150
C *** READ MASTER CONTRCL CARDS                                          CEN05160
C *** NUMNP = TOTAL NUMBER OF NODE POINTS                                CEN05170
C *** NELTYP = NUMBER OF ELEMENT GROUPS                                  CEN05180
      READ(5,9001) NUMNP,(IDOF(I),I=1,6),NEGL,NEGNL,MODEX,NSTE           CEN05190
 9001 FCRMAT(I5,6I1,I4,3I5)                                              CEN05200
      NELTYP=NEGL+NEGNL                                                  CEN05210
```

```
      NNODE=NUMNP
      READ(5,7002) IMASS,IDAMP,IMASSN,ICAMPN
 7002 FORMAT(4I5)
      READ(5,7002) IEIG
      READ(5,7002) ISREF,NUMREF,IEQUIT,ITEMAX
      READ(5,9000) DUMMY
      READ(5,9000) DUMMY
      READ(5,9000) DUMMY
C *** READ OR GENERATE NODAL POINT DATA
      NOLD=0
      NEQ=0
   10 READ(5,9006)CT,N,(ID(I),I=1,6),XPT(N),YPT(N),ZPT(N),KN
 9006 FORMAT(A1,I4,1X,I4,5I5,3F10.0,I5)
C *** CHECK FOR CYLINDRICAL COORDINATES
      IF(CT.NE.CTEST) GO TO 12
          DUM=ZPT(N)/57.2958
          R=YPT(N)
          YPT(N)=R*DCOS(ZPT(N)/57.2958D0)
          ZPT(N)=R*DSIN(ZPT(N)/57.2958D0)
   12 CONTINUE
      NUMPT(N)=N
      IF(NOLD.EQ.0) GO TO 50
C *** FOR GENERATION OF FIXED BOUNDARY CONDITIONS
      DO 15 I=1,6
      IF(IDOLD(I).EQ.-1.AND.ID(I).EQ.0) ID(I)=IDOLD(I)
   15     CONTINUE
      IF(KNOLD.EQ.0) GO TO 50
      NUM=(N-NOLD)/KNOLD
      NUMN=NUM-1
      IF(NUMN.LT.1) GO TO 50
C *** TO COUNT DOFS TO DETERMINE NUMBER OF IC CARDS
      DO 20 I=1,6
          IF(IDOF(I).EQ.0.AND.IDOLD(I).EQ.0) NEQ=NEQ+NUMN
   20 CONTINUE
      DX=(XPT(N)-XPT(NOLD))/NUM
      IF(CT.NE.CTEST) GO TO 21
      ROLD=YPT(NOLD)/DCOS(DUMOLD)
      RNEW=YPT(N)/DCOS(DUM)
      DR=(RNEW-ROLD)/NUM
      DT=(DUM-DUMOLD)/NUM
      GO TO 22
   21 CONTINUE
      DY=(YPT(N)-YPT(NOLD))/NUM
      DZ=(ZPT(N)-ZPT(NOLD))/NUM
   22 CONTINUE
      K=NOLD
      DO 30 J=1,NUMN
        KK=K
```

```
      K=K+KNOLD                                                         CEN05700
      XPT(K)=XPT(KK)+DX                                                 CEN05710
       IF(CT.NE.CTEST) GO TO 26                                         CEN05720
       ROLD=ROLD+DR                                                     CEN05730
       DUMOLD=DUMOLD+DT                                                 CEN05740
       YPT(K)=ROLD*DCOS(DUMOLD)                                         CEN05750
       ZPT(K)=ROLD*DSIN(DUMOLD)                                         CEN05760
       GO TO 28                                                         CEN05770
   26  CONTINUE                                                         CEN05780
      YPT(K)=YPT(KK)+DY                                                 CEN05790
      ZPT(K)=ZPT(KK)+DZ                                                 CEN05800
   28  CONTINUE                                                         CEN05810
      NUMPT(K)=K                                                        CEN05820
   30 CONTINUE                                                          CEN05830
   50 NOLD=N                                                            CEN05840
      KNOLD=KN                                                          CEN05850
      DUMOLD=DUM                                                        CEN05860
C *** TO COUNT DOFS TO DETERMINE NUMBER OF IC CARDS                      CEN05870
      DO 55 I=1,6                                                       CEN05880
       IF(IDOF(I).EQ.0.AND.ID(I).EQ.0) NEQ=NEQ+1                        CEN05890
       IDOLD(I)=ID(I)                                                   CEN05900
   55 CONTINUE                                                          CEN05910
      IF(N.NE.NUMNP) GO TO 10                                           CEN05920
C *** READ LOAD CONTROL CARDS                                           CEN05930
      READ(5,9012)NLOAD,NLCUR,NPTM,IDGRAV                               CEN05940
 9012 FORMAT(4I5)                                                       CEN05950
      DO 80 I=1,IMASSN                                                  CEN05960
      IF(IMASSN.EQ.0) GO TO 81                                          CEN05970
         READ(5,9000) DUMMY                                             CEN05980
   80    CONTINUE                                                       CEN05990
   81 CONTINUE                                                          CEN06000
      IF(IDAMPN.EQ.0) GO TO 91                                          CEN06010
      DO 90 I=1,IDAMPN                                                  CEN06020
         READ(5,9000) DUMMY                                             CEN06030
   90    CONTINUE                                                       CEN06040
   91 CONTINUE                                                          CEN06050
C *** READ INITIAL CONDITIONS                                           CEN06060
      READ(5,9002) ICON                                                 CEN06070
      IF(ICON.EQ.0) GO TO 100                                           CEN06080
      CARDNR=NEQ/6.0                                                    CEN06090
      NCARD=IDINT(CARDNR)                                               CEN06100
      TEST=CARDNR-NCARD                                                 CEN06110
      IF(TEST.GT.0.1) NCARD=NCARD+1                                     CEN06120
      DO 95 I=1,NCARD                                                   CEN06130
         READ(5,9000) DUMMY                                             CEN06140
   95    CONTINUE                                                       CEN06150
      IF(IMASS.EQ.0) GO TO 100                                          CEN06160
      DO 96 I=1,NCARD                                                   CEN06170
```

```
      READ(5,9000) DUMMY                                                CEN06180
   96 CONTINUE                                                          CEN06190
      DO 98 I=1,NCARD                                                   CEN06200
      READ(5,9000) DUMMY                                                CEN06210
   98 CONTINUE                                                          CEN06220
 9007 FORMAT(6E12.6)                                                    CEN06230
  100 CONTINUE                                                          CEN06240
      NUMEL=0                                                           CEN06250
      WRITE(6,9009) NEQ,NCARD                                           CEN06260
 9009 FORMAT(///,' NEQ AND NCARD FOR IC IN GEOM1  = ',I5,10X,I5///)    CEN06270
C *** READ ELEMENT CONTROL CARDS                                         CEN06280
      DO 900 M=1,NELTYP                                                 CEN06290
         READ(5,9008)(NPAR(I),I=1,20)                                   CEN06300
      WRITE(6,9010) (NPAR(I),I=1,20)                                    CEN06310
 9008 FORMAT(20I4)                                                      CEN06320
 9010 FORMAT(///,'  NPAR = ',20I5///)                                   CEN06330
         MTYPE=NPAR(1)                                                  CEN06340
      NUMMAT=NPAR(16)                                                   CEN06350
      NSTRES=NPAR(13)                                                   CEN06360
C *** CALCULATE THE NUMBER OF MATERIAL CASE CARDS                        CEN06370
      IF(NPAR(15).EQ. 1) NCARD=1                                        CEN06380
      IF(NPAR(15).EQ. 2) NCARD=2+NPAR(18)                               CEN06390
      IF(NPAR(15).EQ. 3) NCARD=4                                        CEN06400
      IF(NPAR(15).EQ. 4) NCARD=4                                        CEN06410
      IF(NPAR(15).EQ. 5) NCARD=2                                        CEN06420
      IF(NPAR(15).EQ. 8) NCARD=1                                        CEN06430
      IF(NPAR(15).EQ. 9) NCARD=1                                        CEN06440
      IF(NPAR(15).EQ.10) NCARD=6                                        CEN06450
      IF(NPAR(15).EQ.11) NCARD=6                                        CEN06460
      IF(NPAR(15).NE.12) GO TO 111                                      CEN06470
      CARDNR=NPAR(17)/8.0                                               CEN06480
      NCARD=IDINT(CARDNR)                                               CEN06490
      TEST=CARDNR-NCARD                                                 CEN06500
      IF(TEST.GT.0.1) NCARD=NCARD+1                                     CEN06510
  111 CONTINUE                                                          CEN06520
      N20=20                                                            CEN06530
C *** READ MATERIAL PROPERTIES                                           CEN06540
      DO 222 J=1,NUMMAT                                                 CEN06550
         READ(5,9000) DUMMY                                             CEN06560
 9000 FORMAT(20A4)                                                      CEN06570
         DO 45 I=1,NCARD                                                CEN06580
            READ(5,9000) DUMMY                                          CEN06590
   45       CONTINUE                                                    CEN06600
  222 CONTINUE                                                          CEN06610
C *** READ STRESS OUTPUT TABLE CARDS                                     CEN06620
      IF(NPAR(13).EQ.0) GO TO 61                                        CEN06630
      DO 60 I=1,NSTRES                                                  CEN06640
         READ(5,9000) DUMMY                                             CEN06650
```

```
 6C       CONTINUE                                                     CEN06660
 61 CCNTINUE                                                           CEN06670
    IF(NPAR(14).EQ.0) NPAR(14)=1                                       CEN0668C
    NEL=NPAR(14)-1                                                     CEN06690
130 READ(5,9002) INEL,IINC                                             CEN06700
9002 FCRMAT(I5,30X,I5)                                                 CEN06710
    IF(IINC.EQ.C) IINC=1                                               CEN06720
    READ(5,9004) (INP(I),I=1,8)                                        CEN0673C
    READ(5,9004) (INP(I),I=9,N20)                                      CEN06740
9004 FCRMAT(12I5)                                                      CEN06750
140 NEL=NEL+1                                                          CEN06760
    NREL(NEL)=NEL                                                      CEN06770
    ML=INEL-NEL                                                        CEN06780
    IF(ML) 150,155,160                                                 CEN06790
150 WRITE(6,800C)NEL                                                   CEN06800
800C FCRMAT(//5X,'WARNING - ERROR IN ELEMENT ',I5//)                   CEN06810
C *** NO GENERATICN OF NCDE POINTS REQUIRED                            CEN06820
155 DC 156 I=1,N20                                                     CEN06830
        NP(I)=INP(I)                                                   CEN06840
        NCONT(I,NEL)=NP(I)                                             CEN06850
156     CONTINUE                                                       CEN06860
    GC TO 162                                                          CEN06870
C *** GENERATION CF NODE PCINTS REQUIREC                               CEN06880
160 DC 161 I=1,N20                                                     CEN06890
        IF(NP(I).EQ.0) GO TO 161                                       CEN06900
        NP(I)=NP(I)+KN                                                 CEN06910
        NCONT(I,NEL)=NP(I)                                             CEN06920
161     CONTINUE                                                       CEN06930
162 CCNTINUE                                                           CEN06940
    NUMEL=NUMEL+1                                                      CEN06950
    IF(NEL.EQ.NPAR(2)) RETURN                                          CEN06960
    IF(NEL.LT.INEL) GO TO 140                                          CEN06970
    KN=IINC                                                            CEN06980
    GO TO 130                                                          CEN06990
900     CONTINUE                                                       CEN0700C
    END                                                                CEN07010
```

APPENDIX D

PROGRAM KCONT LISTING

```
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C      *
C      *  PROGRAM KCONT
C      *
C      *  THIS PROGRAM READS AN 8-20 NODE ADINA INPUT DECK AND
C      *  DETERMINES THE UNIDENTIFIED NODE LOCATIONS TO YIELD
C      *  THE COORDINATES AND CONNECTIVITY FOR A 27 NODE BRICK
C      *  MESH.  THE OUTPUT IS PRINTED ON THE LINE PRINTER
C      *  AND WRITTEN ON USER DEFINED FILES 58 AND 59.
C      *
C      *  FILES:
C      *     FT58F001-COORDINATES OF ALL NODES WRITTEN IN
C      *        SEQUENCE AS FOLLOWS:  NODE NUMBER,
C      *        X-COORDINATE,Y-COORDINATE,Z-COORDINATE.
C      *
C      *     FT59F001-CONNECTIVITY ARRAY DIMENSIONED 27 BY
C      *        NUMEL.  STORED IN SEQUENCE BY ROWS.
C      *
C      *  INPUT DATA REQUIREMENTS:
C      *
C      *  CARD 1
C      *    VARIABLE     COLUMN     REMARKS
C      *
C      *     NUMNP        1-10       NUMBER OF NODES IN INPUT
C      *                             MESH.
C      *     NUMEL       11-20       NUMBER OF ELEMENTS IN INPUT
C      *                             MESH
C      *  PLACE HERE THE ADINA INPUT DECK FROM THE TITLE CARD
C      *  TO THE LAST CONNECTIVITY CARD INCLUSIVE.
C      *
C      *  WRITTEN BY:  L.R. EASTERLING NPS, MONTEREY CA., FEB 1978
C      *
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       COMMON A(30000)
       DATA A/30000*0/
       COMMON/IPARA/NUMNP,NUMEL
       READ(5,2000)NUMNP,NUMEL
  2000 FORMAT(2I10)
       MTOT=30000
       N1=1
```

```
      N2=N1+27*NUMEL*2                                                  EAS00420
      N3=N2+27*NUMEL*2                                                  EAS00430
      N4=N3+27*NUMEL*2                                                  EAS00440
      N5=N4+20*NUMEL                                                    EAS00450
      N6=N5+27*NUMEL                                                    EAS00460
      N7=N6+27*NUMEL*2                                                  EAS00470
      N8=N7+NUMEL                                                       EAS00480
      N9=N8+6*NUMEL                                                     EAS00490
      NLAST=N9+12*NUMEL                                                 EAS00500
      IF(NLAST.GT.MTOT)GO TO 10                                         EAS00510
      CALL TOTAL(A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),       EAS00520
     1 A(N9))                                                           EAS00530
      STOP                                                              EAS00540
   10 WRITE(6,1000)NLAST                                                EAS00550
 1000 FORMAT(//5X,'MTOT SHOULD BE INCREASED TO A VALUE OF ',I5//)       EAS00560
      STOP                                                              EAS00570
      END                                                               EAS00580
      SUBROUTINE TOTAL(XPT,YPT,ZPT,NCONT,KCONT,NUMPT,NREL,NFACE,        EAS00590
     1 NLINE)                                                           EAS00600
      IMPLICIT REAL*8(A-H,O-Z)                                          EAS00610
      DIMENSION NLINE(12,1),NFACE(6,1)                                  EAS00620
      COMMON/IPARA/NUMNP,NUMEL                                          EAS00630
      DIMENSION XPT(1),YPT(1),ZPT(1),NCONT(20,1),KCONT(27,1),           EAS00640
     1 NUMPT(1),NREL(1)                                                 EAS00650
      DIMENSION NCON(20),COD(20,3)                                      EAS00660
      DIMENSION NORD(19),IFACE(4,6),ILINE(2,12)                         EAS00670
      DIMENSION CNAT(3,19)                                              EAS00680
      DATA NORD/9,10,11,12,13,14,15,16,17,18,19,20,21,22,               EAS00690
     123,24,25,26,27/                                                   EAS00700
      DATA IFACE/1,2,6,5,2,3,7,6,3,4,8,7,1,4,8,5,1,2,3,4,5,6,7,8/       EAS00710
      DATA ILINE/1,2,2,3,3,4,4,1,5,6,6,7,7,8,8,5,1,5,2,6,3,7,4,8/       EAS00720
      DATA CNAT/0.0D0,1.0D0,1.0D0,-1.0D0,0.0D0,1.0D0,                   EAS00730
     10.0D0,-1.0D0,1.0D0,1.0D0,0.0D0,1.0D0,0.0D0,1.0D0,-1.0D0,          EAS00740
     2-1.0D0,0.0D0,-1.0D0,0.0D0,-1.0D0,-1.0D0,1.0D0,0.0D0,-1.0D0,       EAS00750
     31.0D0,1.0D0,0.0D0,-1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,0.0D0,         EAS00760
     41.0D0,-1.0D0,0.0D0,0.0D0,0.0D0,0.0D0,0.0D0,1.0D0,0.0D0,           EAS00770
     5-1.0D0,0.0D0,0.0D0,0.0D0,-1.0D0,0.0D0,1.0D0,0.0D0,0.0D0,          EAS00780
     60.0D0,0.0D0,1.0D0,0.0D0,0.0D0,-1.0D0/                             EAS00790
      CALL RDADIN(XPT,YPT,ZPT,NCONT,NREL,NUMPT)                         EAS00800
      DO 10 I=1,NUMEL                                                   EAS00810
      DO 10 J=1,27                                                      EAS00820
   10 KCONT(J,I)=0                                                      EAS00830
      DO 20 I=1,NUMEL                                                   EAS00840
      DO 20 J=1,20                                                      EAS00850
   20 KCONT(J,I)=NCONT(J,I)                                             EAS00860
      R=0.0D0                                                           EAS00870
      S=0.0D0                                                           EAS00880
      T=0.0D0                                                           EAS00890
```

```
      NN2=NUMNP                                                         EAS00900
      DO 30 I=1,NUMEL                                                   EAS00910
      NN2=NN2+1                                                         EAS00920
      DO 40 J=1,20                                                      EAS00930
      NCON(J)=NCONT(J,I)                                                EAS00940
      K=NCON(J)                                                         EAS00950
      COD(J,1)=0.0D0                                                    EAS00960
      COD(J,2)=0.0D0                                                    EAS00970
      COD(J,3)=0.0D0                                                    EAS00980
      IF(K.EQ.0)GO TO 40                                                EAS00990
      COD(J,1)=XPT(K)                                                   EAS01000
      COD(J,2)=YPT(K)                                                   EAS01010
      COD(J,3)=ZPT(K)                                                   EAS01020
   40 CONTINUE                                                          EAS01030
      CALL ADCRD(R,S,T,COD,NCON,I,XX,YY,ZZ)                             EAS01040
      XPT(NN2)=XX                                                       EAS01050
      YPT(NN2)=YY                                                       EAS01060
      ZPT(NN2)=ZZ                                                       EAS01070
   30 KCONT(21,I)=NN2                                                   EAS01080
      DO 50 I=1,NUMEL                                                   EAS01090
      DO 50 J=1,6                                                       EAS01100
      NFACE(J,I)=0.0D0                                                  EAS01110
      DO 50 K=1,4                                                       EAS01120
      L=IFACE(K,J)                                                      EAS01130
   50 NFACE(J,I)=NFACE(J,I)+KCONT(L,I)                                  EAS01140
      DO 3000 I=1,NUMEL                                                 EAS01150
      DO 3000 J=1,12                                                    EAS01160
      I1=ILINE(1,J)                                                     EAS01170
      I2=ILINE(2,J)                                                     EAS01180
 3000 NLINE(J,I)=KCONT(I1,I)+KCONT(I2,I)                                EAS01190
      DO 70 I=1,NUMEL                                                   EAS01200
      DO 70 J=1,19                                                      EAS01210
      K=NORD(J)                                                         EAS01220
      IF(KCONT(K,I).NE.0) GO TO 70                                      EAS01230
      NN2=NN2+1                                                         EAS01240
      R=CNAT(1,J)                                                       EAS01250
      S=CNAT(2,J)                                                       EAS01260
      T=CNAT(3,J)                                                       EAS01270
      DO 80 M=1,20                                                      EAS01280
      COD(M,1)=0.0D0                                                    EAS01290
      COD(M,2)=0.0D0                                                    EAS01300
      COD(M,3)=0.0D0                                                    EAS01310
      NCON(M)=NCONT(M,I)                                                EAS01320
      N=NCON(M)                                                         EAS01330
      IF(N.EQ.0)GO TO 80                                                EAS01340
      COD(M,1)=XPT(N)                                                   EAS01350
      COD(M,2)=YPT(N)                                                   EAS01360
      COD(M,3)=ZPT(N)                                                   EAS01370
```

```
   8C CCNTINUE                                                          EAS01380
      CALL ADCRD(R,S,T,COD,NCON,I,XX,YY,ZZ)                             EAS01390
      XFT(NN2)=XX                                                       EAS01400
      YPT(NN2)=YY                                                       EAS01410
      ZPT(NN2)=ZZ                                                       EAS01420
      IF(K.LE.20)GO TO 5000                                             EAS01430
      DC 90 II=1,NUMEL                                                  EAS01440
      DC 90 JJ=1,6                                                      EAS01450
      KK=NORD(JJ+13)                                                    EAS01460
      IF(NFACE(JJ,II).NE.NFACE(J-13,I))GC TO 90                         EAS01470
      KTR=0                                                             EAS01480
      DC 111 L=1,4                                                      EAS01490
      LL=IFACE(L,J-13)                                                  EAS01500
      DC 111 MM=1,8                                                     EAS01510
      IF(KCONT(MM,II).EQ.KCCNT(LL,I))KTR=KTR+1                          EAS01520
  111 CONTINUE                                                          EAS01530
      IF(KTR.EQ.4)KCONT(KK,II)=NN2                                      EAS01540
   90 CCNTINUE                                                          EAS01550
      GC TO 70                                                          EAS01560
 5000 CONTINUE                                                          EAS01570
      DC 4000 II=1,NUMEL                                                EAS01580
      DC 4000 J5=1,12                                                   EAS01590
      KK=NORD(J5)                                                       EAS01600
      IF(NLINE(J5,II).NE.NLINE(J,I))GC TC 4000                          EAS01610
      KTR=0                                                             EAS01620
      DO 4001 L=1,2                                                     EAS01630
      LL=ILINE(L,J)                                                     EAS01640
      DO 4001 MM=1,8                                                    EAS01650
      IF(KCONT(MM,II).EQ.KCCNT(LL,I))KTR=KTR+1                          EAS01660
 4001 CCNTINUE                                                          EAS01670
      IF(KTR.EQ.2)KCONT(KK,II)=NN2                                      EAS01680
 4000 CCNTINUE                                                          EAS01690
   70 CCNTINUE                                                          EAS01700
      WRITE(6,1003)NN2                                                  EAS01710
 1003 FCRMAT(//5X,I7,' CISTINCT POINTS ARE IDENTIFIED'//)               EAS01720
C     REWIND 58                                                         EAS01730
C     REWIND 59                                                         EAS01740
C     WRITE(58)NN2                                                      EAS01750
      WRITE(6,1000)                                                     EAS01760
      DC 100 I=1,NUMEL                                                  EAS01770
      WRITE(6,1001)I                                                    EAS01780
      DC 100 J=1,27                                                     EAS01790
      K=KCONT(J,I)                                                      EAS01800
  100 WRITE(6,1002)J,K,XPT(K),YPT(K),ZPT(K)                             EAS01810
      WRITE(6,1004)                                                     EAS01820
 1004 FCRMAT(//5X,'NODE COORDINATES BY NODE NR'//5X,'NODE',5X,'         EAS01830
     1COORDINATES'/)                                                    EAS01840
      DC 666 I=1,NN2                                                    EAS01850
```

```
  666 WRITE(6,1005)I,XPT(I),YPT(I),ZPT(I)                               EAS01860
C 666 WRITE(58)I,XPT(I),YPT(I),ZPT(I)                                   EAS01870
C     WRITE(59)((KCONT(I,J),J=1,NUMEL),I=1,27)                          EAS01880
 1005 FORMAT(1X,I9,3F16.5)                                              EAS01890
 1000 FORMAT(////5X,'NODE COORDINATES PLUS CENTROIC AND MID-FAC COORDINAEAS01900
     1TES BY ELEMENT',////)                                             EAS01910
 1001 FORMAT(/5X,'ELEMENT ',I5//5X,'NATURAL',5X,'GLOBAL'/5X,'COORD NR', EAS01920
     1 4X,'COORD NR',8X,'X-COORD',8X,'YCOORD',8X,'Z-COORD'/)            EAS01930
 1002 FORMAT(5X,2I10,3F15.5)                                            EAS01940
C     ENDFILE 59                                                        EAS01950
C     ENDFILE 58                                                        EAS01960
      RETURN                                                            EAS01970
      END                                                               EAS01980
      SUBROUTINE ADCRD(R,S,T,COD,NCON,IEL,XX,YY,ZZ)                     EAS01990
      IMPLICIT REAL*8(A-H,O-Z)                                          EAS02000
      DIMENSION EMI(20)                                                 EAS02010
      DIMENSION COD(20,3),PROD1(3)                                      EAS02020
      DIMENSION NCON(20)                                                EAS02030
      GP(BT) = 0.5D0*(1.0D0 + BT)                                       EAS02040
      GM(BT) = 0.5D0*(1.0D0 - BT)                                       EAS02050
      GZ(BT) = 1.0D0 - BT*BT                                            EAS02060
      ZRO = 0.0D0                                                       EAS02070
      ONE = 1.0D0                                                       EAS02080
      TWO = 2.0D0                                                       EAS02090
      EMI( 9) = GZ(R)*GP(S)*GP(T)                                       EAS02100
      IF(NCON( 9).EQ.0) EMI( 9) = ZRO                                   EAS02110
      EMI(10) = GM(R)*GZ(S)*GP(T)                                       EAS02120
      IF(NCON(10).EQ.0) EMI(10) = ZRO                                   EAS02130
      EMI(11) = GZ(R)*GM(S)*GP(T)                                       EAS02140
      IF(NCON(11).EQ.0) EMI(11) = ZRO                                   EAS02150
      EMI(12) = GP(R)*GZ(S)*GP(T)                                       EAS02160
      IF(NCON(12).EQ.0) EMI(12) = ZRO                                   EAS02170
      EMI(13) = GZ(R)*GP(S)*GM(T)                                       EAS02180
      IF(NCON(13).EQ.0) EMI(13) = ZRO                                   EAS02190
      EMI(14) = GM(R)*GZ(S)*GM(T)                                       EAS02200
      IF(NCON(14).EQ.0) EMI(14) = ZRO                                   EAS02210
      EMI(15) = GZ(R)*GM(S)*GM(T)                                       EAS02220
      IF(NCON(15).EQ.0) EMI(15) = ZRO                                   EAS02230
      EMI(16) = GP(R)*GZ(S)*GM(T)                                       EAS02240
      IF(NCON(16).EQ.0) EMI(16) = ZRO                                   EAS02250
      EMI(17) = GP(R)*GP(S)*GZ(T)                                       EAS02260
      IF(NCON(17).EQ.0) EMI(17) = ZRO                                   EAS02270
      EMI(18) = GM(R)*GP(S)*GZ(T)                                       EAS02280
      IF(NCON(18).EQ.0) EMI(18) = ZRO                                   EAS02290
      EMI(19) = GM(R)*GM(S)*GZ(T)                                       EAS02300
      IF(NCON(19).EQ.0) EMI(19) = ZRO                                   EAS02310
      EMI(20) = GP(R)*GM(S)*GZ(T)                                       EAS02320
      IF(NCON(20).EQ.0) EMI(20) = ZRO                                   EAS02330
```

```
      EMI( 1) = GP(R)*GP(S)*GP(T) - (EMI( 9) + EMI(12) + EMI(17))/TWO    EAS02340
      EMI( 2) = GM(R)*GP(S)*GP(T) - (EMI( 9) + EMI(10) + EMI(18))/TWO    EAS02350
      EMI( 3) = GM(R)*GM(S)*GP(T) - (EMI(10) + EMI(11) + EMI(19))/TWO    EAS02360
      EMI( 4) = GP(R)*GM(S)*GP(T) - (EMI(11) + EMI(12) + EMI(20))/TWO    EAS02370
      EMI( 5) = GP(R)*GP(S)*GM(T) - (EMI(13) + EMI(16) + EMI(17))/TWO    EAS02380
      EMI( 6) = GM(R)*GP(S)*GM(T) - (EMI(13) + EMI(14) + EMI(18))/TWO    EAS02390
      EMI( 7) = GM(R)*GM(S)*GM(T) - (EMI(14) + EMI(15) + EMI(19))/TWO    EAS02400
      EMI( 8) = GP(R)*GM(S)*GM(T) - (EMI(15) + EMI(16) + EMI(20))/TWO    EAS02410
      DO 700 I=1,3                                                       EAS02420
      PROD1(I)=0.0D0                                                     EAS02430
      DO 700 J=1,20                                                      EAS02440
  700 PROD1(I)=PROD1(I)+EMI(J)*COD(J,I)                                  EAS02450
      XX=PROD1(1)                                                        EAS02460
      YY=PROD1(2)                                                        EAS02470
      ZZ=PROD1(3)                                                        EAS02480
      RETURN                                                             EAS02490
      END                                                                EAS02500
      SUBROUTINE RDADIN(XPT,YPT,ZPT,NCCNT,NREL,NUMPT)                    EAS02510
C                                                                        EAS02520
C     ************************************************************       EAS02530
C                                                                        EAS02540
C        SUBROUTINE TO READ ADINA DECK.                                  EAS02550
C                                                                        EAS02560
C     ************************************************************       EAS02570
C                                                                        EAS02580
      IMPLICIT REAL*8 (A-H,O-Z)                                          EAS02590
      REAL*4 TITLE(20)                                                   EAS02600
      COMMON/IPARA/NUMNP,NUMEL                                           EAS02610
      COMMON/LOADS/NLOAD,NLCUR,NPTM,ICGRAV                               EAS02620
      DIMENSION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1),NCCNT(20,1)        EAS02630
      DIMENSION IDOF(6),ID(6),IDOLD(6),NCDE(20),NPAR(20)                 EAS02640
     1 ,NP(20),INP(20)                                                   EAS02650
      DATA CTEST/'X       '/                                             EAS02660
      NCARD=0                                                            EAS02670
      READ(5,9000)(TITLE(I),I=1,20)                                      EAS02680
      WRITE(6,9011)(TITLE(I),I=1,20)                                     EAS02690
 9011 FORMAT(///5X,20A4,///)                                             EAS02700
C *** READ MASTER CONTROL CARDS                                          EAS02710
C *** NUMNP = TOTAL NUMBER OF NODE POINTS                                EAS02720
C *** NELTYP = NUMBER OF ELEMENT GROUPS                                  EAS02730
      READ(5,9001) NUMNP,(IDOF(I),I=1,6),NEGL,NEGNL,MODEX,NSTE           EAS02740
 9001 FORMAT(I5,6I1,I4,3I5)                                              EAS02750
      NELTYP=NEGL+NEGNL                                                  EAS02760
      NNODE=NUMNP                                                        EAS02770
      READ(5,7002) IMASS,IDAMP,IMASSN,IDAMPN                             EAS02780
 7002 FORMAT(4I5)                                                        EAS02790
      READ(5,7002) IEIG                                                  EAS02800
      READ(5,7002) ISREF,NUMREF,IEQUIT,ITEMAX                            EAS02810
```

```
      READ(5,9000) DUMMY                                               EAS02820
      READ(5,9000) DUMMY                                               EAS02830
      READ(5,9000) DUMMY                                               EAS02840
C *** READ OR GENERATE NODAL POINT DATA                                 EAS02850
      NCLD=0                                                           EAS02860
      NEQ=0                                                            EAS02870
   10 READ(5,9006)CT,N,(ID(I),I=1,6),XPT(N),YPT(N),ZPT(N),KN           EAS02880
 9006 FCRMAT(A1,I4,1X,I4,5I5,3F10.0,I5)                                EAS02890
C *** CHECK FOR CYLINDRICAL COORDINATES                                 EAS02900
      IF(CT.NE.CTEST) GC TO 12                                         EAS02910
            DUM=ZPT(N)/57.2958                                         EAS02920
            R=YPT(N)                                                   EAS02930
            YPT(N)=R*DCOS(ZPT(N)/57.2958D0)                            EAS02940
            ZPT(N)=R*DSIN(ZPT(N)/57.2958D0)                            EAS02950
   12 CONTINUE                                                         EAS02960
      NUMPT(N)=N                                                       EAS02970
      IF(NOLD.EQ.0) GO TC 50                                           EAS02980
C *** FOR GENERATION OF FIXED BOUNDARY CONDITIONS                       EAS02990
      DC 15 I=1,6                                                      EAS03000
      IF(IDOLD(I).EQ.-1.AND.ID(I).EQ.0) ID(I)=IDOLD(I)                 EAS03010
   15       CONTINUE                                                   EAS03020
      IF(KNOLD.EQ.0) GO TO 50                                          EAS03030
      NUM=(N-NOLD)/KNOLD                                               EAS03040
      NUMN=NUM-1                                                       EAS03050
      IF(NUMN.LT.1) GO TC 50                                           EAS03060
C *** TO COUNT DOFS TO DETERMINE NUMBER OF IC CARDS                     EAS03070
      DC 20 I=1,6                                                      EAS03080
            IF(IDOF(I).EQ.0.AND.IDOLD(I).EQ.0) NEQ=NEQ+NUMN            EAS03090
   20 CONTINUE                                                         EAS03100
      DX=(XPT(N)-XPT(NOLD))/NUM                                        EAS03110
      IF(CT.NE.CTEST) GO TO 21                                         EAS03120
      ROLD=YPT(NOLD)/DCOS(DUMOLD)                                      EAS03130
      RNEW=YPT(N)/DCOS(DUM)                                            EAS03140
      DR=(RNEW-ROLD)/NUM                                               EAS03150
      DT=(DUM-DUMOLD)/NUM                                              EAS03160
      GC TO 22                                                         EAS03170
   21 CONTINUE                                                         EAS03180
      DY=(YPT(N)-YPT(NOLD))/NUM                                        EAS03190
      DZ=(ZPT(N)-ZPT(NOLD))/NUM                                        EAS03200
   22 CONTINUE                                                         EAS03210
      K=NOLD                                                           EAS03220
      DC 30 J=1,NUMN                                                   EAS03230
           KK=K                                                        EAS03240
           K=K+KNOLD                                                   EAS03250
           XPT(K)=XPT(KK)+DX                                           EAS03260
            IF(CT.NE.CTEST) GO TO 26                                   EAS03270
            ROLD=ROLD+DR                                               EAS03280
            DUMOLD=DUMOLD+DT                                           EAS03290
```

```
            YPT(K)=ROLD*DCOS(DUMOLD)                                      EAS03300
            ZPT(K)=ROLD*DSIN(DUMOLD)                                      EAS03310
            GO TO 28                                                      EAS03320
   26       CONTINUE                                                      EAS03330
           YPT(K)=YPT(KK)+DY                                              EAS03340
           ZPT(K)=ZPT(KK)+DZ                                              EAS03350
   28       CONTINUE                                                      EAS03360
           NUMPT(K)=K                                                     EAS03370
   30      CONTINUE                                                       EAS03380
   50 NOLD=N                                                              EAS03390
      KNOLD=KN                                                            EAS03400
      DUMOLD=DUM                                                          EAS03410
C *** TO COUNT DOFS TO DETERMINE NUMBER OF IC CARDS                        EAS03420
      DO 55 I=1,6                                                         EAS03430
            IF(IDOF(I).EQ.0.AND.ID(I).EQ.0) NEQ=NEQ+1                     EAS03440
            IDOLD(I)=ID(I)                                                EAS03450
   55      CONTINUE                                                       EAS03460
      IF(N.NE.NUMNP) GO TO 10                                             EAS03470
C *** READ LOAD CONTROL CARDS                                             EAS03480
      READ(5,9012)NLOAD,NLCUR,NPTM,IDGRAV                                 EAS03490
 9012 FORMAT(4I5)                                                         EAS03500
      DO 80 I=1,IMASSN                                                    EAS03510
      IF(IMASSN.EQ.0) GO TO 81                                            EAS03520
           READ(5,9000) DUMMY                                             EAS03530
   80      CONTINUE                                                       EAS03540
   81 CONTINUE                                                            EAS03550
      IF(IDAMPN.EQ.0) GO TO 91                                            EAS03560
      DO 90 I=1,IDAMPN                                                    EAS03570
           READ(5,9000) DUMMY                                             EAS03580
   90      CONTINUE                                                       EAS03590
   91 CONTINUE                                                            EAS03600
C *** READ INITIAL CONDITIONS                                             EAS03610
      READ(5,9002) ICON                                                   EAS03620
      IF(ICON.EQ.0) GO TO 100                                             EAS03630
      CARDNR=NEQ/6.0                                                      EAS03640
      NCARD=IDINT(CARDNR)                                                 EAS03650
      TEST=CARDNR-NCARD                                                   EAS03660
      IF(TEST.GT.0.1) NCARD=NCARD+1                                       EAS03670
      DO 95 I=1,NCARD                                                     EAS03680
           READ(5,9000) DUMMY                                             EAS03690
   95      CONTINUE                                                       EAS03700
      IF(IMASS.EQ.0) GO TO 100                                            EAS03710
      DO 96 I=1,NCARD                                                     EAS03720
           READ(5,9000) DUMMY                                             EAS03730
   96      CONTINUE                                                       EAS03740
      DO 98 I=1,NCARD                                                     EAS03750
           READ(5,9000) DUMMY                                             EAS03760
   98      CONTINUE                                                       EAS03770
```

```
 9007 FORMAT(6E12.6)                                                        EAS03780
  100 CONTINUE                                                              EAS03790
      NUMEL=0                                                               EAS03800
      WRITE(6,9009) NEQ,NCARD                                               EAS03810
 9009 FORMAT(///,' NEQ AND NCARD FOR IC IN GEOM1  = ',I5,10X,I5///)         EAS03820
C *** READ ELEMENT CONTROL CARDS                                             EAS03830
      DO 900 M=1,NELTYP                                                     EAS03840
          READ(5,9008)(NPAR(I),I=1,20)                                      EAS03850
      WRITE(6,9010) (NPAR(I),I=1,20)                                        EAS03860
 9008 FORMAT(20I4)                                                          EAS03870
 9010 FORMAT(///,'  NPAR = ',20I5///)                                       EAS03880
          MTYPE=NPAR(1)                                                     EAS03890
      NUMMAT=NPAR(16)                                                       EAS03900
      NSTRES=NPAR(13)                                                       EAS03910
C *** CALCULATE THE NUMBER OF MATERIAL CASE CARDS                           EAS03920
      IF(NPAR(15).EQ. 1) NCARD=1                                            EAS03930
      IF(NPAR(15).EQ. 2) NCARD=2+NPAR(18)                                   EAS03940
      IF(NPAR(15).EQ. 3) NCARD=4                                            EAS03950
      IF(NPAR(15).EQ. 4) NCARD=4                                            EAS03960
      IF(NPAR(15).EQ. 5) NCARD=2                                            EAS03970
      IF(NPAR(15).EQ. 8) NCARD=1                                            EAS03980
      IF(NPAR(15).EQ. 9) NCARD=1                                            EAS03990
      IF(NPAR(15).EQ.10) NCARD=6                                            EAS04000
      IF(NPAR(15).EQ.11) NCARD=6                                            EAS04010
      IF(NPAR(15).NE.12) GO TO 111                                          EAS04020
      CARDNR=NPAR(17)/8.0                                                   EAS04030
      NCARD=IDINT(CARDNR)                                                   EAS04040
      TEST=CARDNR-NCARD                                                     EAS04050
      IF(TEST.GT.0 1) NCARD=NCARD+1                                         EAS04060
  111 CONTINUE                                                              EAS04070
      N20=20                                                                EAS04080
C *** READ MATERIAL PROPERTIES                                              EAS04090
      DO 222 J=1,NUMMAT                                                     EAS04100
          READ(5,9000) DUMMY                                                EAS04110
 9000 FORMAT(20A4)                                                          EAS04120
          DO 45 I=1,NCARD                                                   EAS04130
              READ(5,9000) DUMMY                                            EAS04140
   45         CONTINUE                                                      EAS04150
  222 CONTINUE                                                              EAS04160
C *** READ STRESS OUTPUT TABLE CARDS                                        EAS04170
      IF(NPAR(13).EQ.0) GO TO 61                                            EAS04180
      DO 60 I=1,NSTRES                                                      EAS04190
          READ(5,9000) DUMMY                                                EAS04200
   60     CONTINUE                                                          EAS04210
   61 CONTINUE                                                              EAS04220
      IF(NPAR(14).EQ.0) NPAR(14)=1                                          EAS04230
      NEL=NPAR(14)-1                                                        EAS04240
  130 READ(5,9002) INEL,IINC                                                EAS04250
```

```
 9002 FORMAT(I5,30X,I5)                                                  EAS04260
      IF(IINC.EQ.0) IINC=1                                               EAS04270
      READ(5,9004) (INP(I),I=1,8)                                        EAS04280
      READ(5,9004) (INP(I),I=9,N20)                                      EAS04290
 9004 FORMAT(12I5)                                                       EAS04300
  140 NEL=NEL+1                                                          EAS04310
      NREL(NEL)=NEL                                                      EAS04320
      ML=INEL-NEL                                                        EAS04330
      IF(ML) 150,155,160                                                 EAS04340
  150 WRITE(6,8000)NEL                                                   EAS04350
 8000 FORMAT(//5X,'WARNING - ERROR IN ELEMENT ',I5//)                    EAS04360
C *** NO GENERATION OF NODE POINTS REQUIRED                               EAS04370
  155 DO 156 I=1,N20                                                     EAS04380
         NP(I)=INP(I)                                                    EAS04390
         NCONT(I,NEL)=NP(I)                                              EAS04400
  156    CONTINUE                                                        EAS04410
      GO TO 162                                                          EAS04420
C *** GENERATION OF NODE POINTS REQUIRED                                  EAS04430
  160 DO 161 I=1,N20                                                     EAS04440
         IF(NP(I).EQ.0) GO TO 161                                        EAS04450
         NP(I)=NP(I)+KN                                                  EAS04460
         NCONT(I,NEL)=NP(I)                                              EAS04470
  161    CONTINUE                                                        EAS04480
  162 CONTINUE                                                           EAS04490
      NUMEL=NUMEL+1                                                      EAS04500
      IF(NEL.EQ.NPAR(2)) RETURN                                          EAS04510
      IF(NEL.LT.INEL) GO TO 140                                          EAS04520
      KN=IINC                                                            EAS04530
      GO TO 130                                                          EAS04540
  900    CONTINUE                                                        EAS04550
      END                                                                EAS04560
```

APPENDIX E

PROGRAM STRESS LISTING

```
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C      *
C      *   PROGRAM STRESS
C      *
C      *   THIS PROGRAM TAKES THE CONNECTIVITY DATA FROM PROGRAM
C      *   KCONT AND THE STRESS RESULTS FROM AN ADINA ANALYSIS WITH
C      *   THE 27 NODE BRICK OUTPUT OPTION USED AND AVERAGES THE
C      *   CONTRIUBTIONS TO EACH NODE BY THE VARIOUS ELEMENTS TO
C      *   WHICH IT IS A MEMBER.  THESE VALUES ARE THEN OPERATED ON
C      *   TO YIELD THE THREE PRINCIPAL STRESSES AND AN ARRAY
C      *   CONTAINING THE MAXIMUM PRINCIPAL STRESS FOR EACH NODE.  THE
C      *   OUTPUT CONSISTS OF A LINE PRINTER LISTING OF AVERAGE AND
C      *   PRINCIPAL STRESSES AND STORAGE ON USER DEFINED FILES 60
C      *   AND 61 RESPECTIVELY.
C      *
C      *   OUTPUT STORAGE:
C      *     FT60F001-SEQUENTIAL LISTING OF THE AVERAGE STRESS ARRAYS
C      *       IN THE FOLLOWING ORDER:  ASIGX,ASIGY,ASIGZ,ATAUXY,
C      *       ATAUXZ,ATAUYZ.
C      *     FT61F001-SEQUENTIAL LISTING OF THE PRINCIPAL STRESS ARRAYS
C      *       IN THE FOLLOWING ORDER:  PSIG1,PSIG2,PSIG3,PMAX.
C      *
C      *   INPUT DATA REQUIREMENTS:
C      *    CARD 1
C      *     VARIABLE   COLUMN    REMARKS
C      *      NN2         1-5       NUMBER OF NODES IN 27 NODE BRICK MESH
C      *      NUMEL       6-10      NUMBER OF ELEMENTS IN MESH
C      *
C      *   DD CARDS ARE REQUIRED FOR FILE 57 ESTABLISHED BY ADINA
C      *   AND FILES 58 AND 59 ESTABLISHED BY PROGRAM KCONT.
C      *
C      *   WRITTEN BY: L.R. EASTERLING NPS, MONTEREY CA., FEB 1978
C      *
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       DIMENSION A(68000)
       DATA A/68000*0./
       MTOT=68000
       READ(5,1000)NN2,NUMEL
       N1=1
```

```
      N2=N1+27*NUMEL*2                                                  STR00410
      N3=N2+27*NUMEL*2                                                  STR00420
      N4=N3+27*NUMEL*2                                                  STR00430
      N5=N4+27*NUMEL*2                                                  STR00440
      N6=N5+27*NUMEL*2                                                  STR00450
      N7=N6+27*NUMEL*2                                                  STR00460
      N8=N7+NN2*2                                                       STR00470
      N9=N8+NN2*2                                                       STR00480
      N10=N9+NN2*2                                                      STR00490
      N11=N10+NN2*2                                                     STR00500
      N12=N11+NN2*2                                                     STR00510
      N13=N12+NN2*2                                                     STR00520
      N14=N13+NN2*2                                                     STR00530
      N15=N14+NN2*2                                                     STR00540
      N16=N15+NN2*2                                                     STR00550
      N17=N16+NN2*2                                                     STR00560
      N18=N17+3*NN2*2                                                   STR00570
      N19=N18+NN2                                                       STR00580
      NLAST=N19+27*NUMEL                                                STR00590
      IF(NLAST.GE.MTOT)GO TO 10                                         STR00600
      CALL STRESS(A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),      STR00610
     1A(N9),A(N10),A(N11),A(N12),A(N13),A(N14),A(N15),A(N16),A(N17),    STR00620
     2A(N18),A(N19),NN2,NUMEL)                                          STR00630
      STOP                                                              STR00640
   10 WRITE(6,1001)NLAST                                                STR00650
 1000 FORMAT(2I5)                                                       STR00660
 1001 FORMAT(///5X,'*****MTOT INSUFFICIENT-MUST BE GREATER THAN',       STR00670
     1I10////)                                                          STR00680
      STOP                                                              STR00690
      END                                                               STR00700
      SUBROUTINE STRESS (SIGX,SIGY,SIGZ,TAUXY,TAUXZ,TAUYZ,ASIGX,ASIGY,ASSTR00710
     1IGZ,ATAUXY,ATAUXZ,ATAUYZ,PSIG1,PSIG2,PSIG3,FMAX,COD,KTR,KCONT,    STR00720
     2NN2,NUMEL)                                                        STR00730
      IMPLICIT REAL*8(A-H,O-Z)                                          STR00740
      DIMENSION SIGX(27,NUMEL),SIGY(27,NUMEL),SIGZ(27,NUMEL),
     1TAUXY(27,NUMEL),TAUXZ(27,NUMEL),TAUYZ(27,NUMEL),ASIGX(NN2),
     2ASIGY(NN2),ASIGZ(NN2),ATAUXY(NN2),ATAUXZ(NN2),ATAUYZ(NN2),
     3PSIG1(NN2),PSIG2(NN2),PSIG3(NN2),FMAX(NN2),COD(3,NN2)
      DIMENSION KTR(1),KCONT(27,1)                                      STR00790
      REWIND 59                                                         STR00800
      REWIND 58                                                         STR00810
      REWIND 57                                                         STR00820
      READ(59)((KCONT(I,J),J=1,NUMEL),I=1,27)                           STR00830
      READ(58)NN2                                                       STR00840
      DO 10 I=1,NUMEL                                                   STR00850
      READ(57)IEL                                                       STR00860
      DO 10 J=1,27                                                      STR00870
      READ(57)NDUM,SIGX(J,IEL),SIGY(J,IEL),SIGZ(J,IEL),TAUXY(J,IEL),    STR00880
```

```
    1 TAUXZ(J,IEL),TAUYZ(J,IEL)                                      STR00890
   10 CONTINUE                                                       STR00900
      DO 20 I=1,NN2                                                  STR00910
   20 READ(58)J,(COD(K,J),K=1,3)                                     STR00920
      DO 30 I=1,NN2                                                  STR00930
      ASIGX(I)=0.0D0                                                 STR00940
      ASIGY(I)=0.0D0                                                 STR00950
      ASIGZ(I)=0.0D0                                                 STR00960
      ATAUXY(I)=0.0D0                                                STR00970
      ATAUXZ(I)=0.0D0                                                STR00980
      ATAUYZ(I)=0.0D0                                                STR00990
   30 KTR(I)=0                                                       STR01000
      DO 40 I=1,NUMEL                                                STR01010
      DO 40 J=1,27                                                   STR01020
      K=KCONT(J,I)                                                   STR01030
      ASIGX(K)=ASIGX(K)+SIGX(J,I)                                    STR01040
      ASIGY(K)=ASIGY(K)+SIGY(J,I)                                    STR01050
      ASIGZ(K)=ASIGZ(K)+SIGZ(J,I)                                    STR01060
      ATAUXY(K)=ATAUXY(K)+TAUXY(J,I)                                 STR01070
      ATAUXZ(K)=ATAUXZ(K)+TAUXZ(J,I)                                 STR01080
      ATAUYZ(K)=ATAUYZ(K)+TAUYZ(J,I)                                 STR01090
   40 KTR(K)=KTR(K)+1                                                STR01100
      DO 50 I=1,NN2                                                  STR01110
      ASIGX(I)=ASIGX(I)/KTR(I)                                       STR01120
      ASIGY(I)=ASIGY(I)/KTR(I)                                       STR01130
      ASIGZ(I)=ASIGZ(I)/KTR(I)                                       STR01140
      ATAUXY(I)=ATAUXY(I)/KTR(I)                                     STR01150
      ATAUXZ(I)=ATAUXZ(I)/KTR(I)                                     STR01160
   50 ATAUYZ(I)=ATAUYZ(I)/KTR(I)                                     STR01170
      REWIND 60                                                      STR01180
      WRITE(60)ASIGX,ASIGY,ASIGZ,ATAUXY,ATAUXZ,ATAUYZ                STR01190
      WRITE(6,1001)                                                  STR01200
 1001 FORMAT(////5X,'AVERAGED NODAL STRESSES'////)                   STR01210
      DO 60 I=1,NN2                                                  STR01220
      WRITE(6,1000)I,ASIGX(I),ASIGY(I),ASIGZ(I),ATAUXY(I),           STR01230
    1 ATAUXZ(I),ATAUYZ(I)                                            STR01240
   60 CONTINUE                                                       STR01250
 1000 FORMAT(1X,I6,6F16.5)                                           STR01260
      OTHRD=1.0D0/3.0D0                                              STR01270
      PI=3.141592654                                                 STR01280
      DO 100 I=1,NN2                                                 STR01290
      P=-(ASIGX(I)+ASIGY(I)+ASIGZ(I))                                STR01300
      Q=ASIGX(I)*ASIGY(I)+ASIGY(I)*ASIGZ(I)+ASIGX(I)*ASIGZ(I)        STR01310
    1 -ATAUXY(I)**2-ATAUXZ(I)**2-ATAUYZ(I)**2                        STR01320
      R=-(ASIGX(I)*ASIGY(I)*ASIGZ(I)+2.0D0*ATAUXY(I)*ATAUXZ(I)*ATAUYZ(I)STR01330
    1 -ASIGX(I)*ATAUYZ(I)**2-ASIGY(I)*ATAUXZ(I)**2-ASIGZ(I)*ATAUXY(I)**STR01340
    22)                                                              STR01350
      A=(3.0D0*Q-P**2)/3.0D0                                         STR01360
```

```
      B=(2.0D0*P**3-9.0D0*P*Q+27.0D0*R)/27.0D0
      AB=(B**2)/4.0D0+(A**3)/27.0D0
      IF(AB)400,300,200
  200 WRITE(6,2000)I,AB,A,B,P,Q,R
      PSIG1(I)=11111111.
      PSIG2(I)=PSIG1(I)
      PSIG3(I)=PSIG1(I)
      GO TO 100
  300 AA=(-B/2.0D0)**OTHRD
      BB=AA
      PSIG1(I)=AA*2.0D0-P/3.0D0
      PMAX(I)=PSIG1(I)
      PSIG2(I)=-AA-P/3.0D0
      PSIG3(I)=PSIG2(I)
      IF(PSIG2(I).GT.PSIG1(I))PMAX(I)=PSIG2(I)
      GO TO 100
  400 PHI=DARCOS((-B/2.0D0)/DSQRT((-A**3)/27.0D0))
      PSIG1(I)=2.0D0*DSQRT(-A/3.0D0)*DCOS(PHI/3.0D0)-P/3.0D0
      PSIG2(I)=2.0D0*DSQRT(-A/3.0D0)*DCOS(PHI/3.0D0+2.0D0*PI/3.0D0)
     1 -P/3.0D0
      PSIG3(I)=2.0D0*DSQRT(-A/3.0D0)*DCOS(PHI/3.0D0+4.0D0*PI/3.0D0)
     1 -P/3.0D0
      PMAX(I)=PSIG1(I)
      IF(PSIG2(I).GT.PMAX(I))PMAX(I)=PSIG2(I)
      IF(PSIG3(I).GT.PMAX(I))PMAX(I)=PSIG3(I)
  100 CONTINUE
      REWIND 61
      WRITE(61)PSIG1,PSIG2,PSIG3,PMAX
      WRITE(6,2001)
      DO 600 I=1,NN2
  600 WRITE(6,2002)I,PMAX(I),PSIG1(I),PSIG2(I),PSIG3(I)
 2000 FORMAT(///5X,'SOLUTION FOR NODE',I5,'IS IMAGINARY'//
     1 10X,'AB= ',G25.16/10X,'A=',G25.16/10X,'B=',G25.16/
     2 10X,'P= ',G25.16/10X,'Q= ',G25.16/10X,'R= ',G25.16///)
 2001 FORMAT(///20X,'NODAL PRINCIPAL STRESS FOR FIRST STAGE ROTOR BLADE'
     1 ///5X,'NODE',5X,'MAXIMUM',13X,'NR 1',16X,'NR 2',16X,'NR 3'//)
 2002 FORMAT(1X,I8,5X,4G20.11)
      ENDFILE 61
      ENDFILE 60
      RETURN
      END
```

APPENDIX F

PROGRAM CONTOUR PLOT DATA LISTING

```
C     **************************************************************CTRP0010
C     *                                                             CTRP0020
C     *                                                             CTRP0030
C     *       PROGRAM CONTOUR PLOT DATA                             CTRP0040
C     *                                                             CTRP0050
C     *       THIS PROGRAM IS DESIGNED TO TAKE NODAL POINT DATA FROM  CTRP0060
C     *       AN ADINA 3-D MESH FROM FILE 58 OF PROGRAM KCONT AND PLOT CTRP0070
C     *       NODE LOCATION MARKS AND NODE NUMBERS FOR VARIOUS DESIRED CTRP0080
C     *       2-DIMENSIONAL PLANES.  PLANES MAY BE DEFINED BY SPECIFIC CTRP0090
C     *       COORDINATE VALUE, LIMITING COORDINATE VALUES, OR INPUT  CTRP0100
C     *       OF DESIRED NODE  NUMBERS TO BE PLOTTED.               CTRP0110
C     *       DATA FOR CONTOUR PLOTTING CAN ALSO BE OUTPUT          CTRP0120
C     *       ON THE LINE PRINTER AND/OR PUNCHED CARDS.             CTRP0130
C     *                                                             CTRP0140
C     *       REQUIRED DATA DECK:                                   CTRP0150
C     *                                                             CTRP0160
C     *       CONTROL CARD 1  (FORMAT 4I10)                         CTRP0170
C     *                                                             CTRP0180
C     *          VARIABLE         COL           REMARKS             CTRP0190
C     *                                                             CTRP0200
C     *          NN2              1-10          TOTAL NUMBER OF NODES  CTRP0210
C     *          NPICT            11-20         TOTAL NUMBER OF       CTRP0220
C     *                                         DIFFERENT PLANES TO BE CTRP0230
C     *                                         PLOTTED               CTRP0240
C     *          NDMAX            21-30         HIGHEST NUMBER OF     CTRP0250
C     *                                         NODES  EXPECTED PER   CTRP0260
C     *                                         PLANE                 CTRP0270
C     *          NPCH             31-40         FLAG FOR PUNCHING     CTRP0280
C     *                                         CONTOUR PLOT DATA     CTRP0290
C     *                                           0 FOR PUNCHING      CTRP0300
C     *                                           1 FOR NO PUNCH      CTRP0310
C     *                                                             CTRP0320
C     *       NODE CARDS                                            CTRP0330
C     *          INPUT NPICT SETS OF CARDS.  LEAVE COLUNMNS FOR     CTRP0340
C     *          TESTL,TESTH,ICRD BLANK IF NODES TO BE READ IN.     CTRP0350
C     *          FOR EQUALITY TEST INPUT TESTL=TESTH.               CTRP0360
C     *                                                             CTRP0370
C     *       CARD1  (FORMAT 20A4)  TITLE CARD FOR ANALYSIS         CTRP0380
C     *                              PLANE.                         CTRP0390
C     *                                                             CTRP0400
```

```
C      *       CARD 2  (FORMAT 2F10,3I5)                                          CTRP0410
C      *                                                                          CTRP0420
C      *                                                                          CTRP0430
C      *          TESTL              1-10          LOWER LIMIT OF PLANE           CTRP0440
C      *          TESTH              11-20         HIGHER LIMIT OF PLANE          CTRP0450
C      *          ICRD               21-25         COORDINATE TO BE               CTRP0460
C      *                                           TESTED                         CTRP0470
C      *                                            1=X COORD                     CTRP0480
C      *                                            2=Y COORD                     CTRP0490
C      *                                            3=Z COORD                     CTRP0500
C      *          IRD                 26-30        FLAG FOR READING IN            CTRP0510
C      *                                           NODE NUMBERS                   CTRP0520
C      *                                            0=NO CARDS READ               CTRP0530
C      *                                            1=READ NODAL PT DATA          CTRP0540
C      *                                               CARDS                      CTRP0550
C      *          ICT                31-35         NR OF NODES TO BE              CTRP0560
C      *                                           READ IN.                       CTRP0570
C      *          MX                  11-15         COORDINATE TO BE              CTRP0580
C      *                                            PLOTTED AS HOR.COMP           CTRP0590
C      *          MY                  16-20         COORDINATE TO BE              CTRP0600
C      *                                            PLOTTED AS VERT.              CTRP0610
C      *          IF IRD=1, PLACE HERE SUFFICIENT CARDS TO DEFINE                 CTRP0620
C      *          NODE NUMBERS TO BE PLOTTED USING FORMAT 16I5                    CTRP0630
C      *                                                                          CTRP0640
C      *       CARD 3-PLOT TITLE CARD FORMAT 20A4 ANY TITLE                       CTRP0650
C      *          TO BE PLOTTED BELOW FIRST PLOT AND ABOVE FINAL                  CTRP0660
C      *          PLOT.  MAXIMUM OF 80 CHARACTERS.                                CTRP0670
C      *                                                                          CTRP0680
C      *       CARD 4 (FORMAT F10.4)                                              CTRP0690
C      *                                                                          CTRP0700
C      *          DMAGS               1-10          MAGNIFICATION OF              CTRP0710
C      *                                            COORDINATE VALUE SO           CTRP0720
C      *                                            THAT MAXIMUM DIMEN-           CTRP0730
C      *                                            SION DOES NOT                 CTRP0740
C      *                                            EXCEED 9 IN.                  CTRP0750
C      *                                             1=X COORD                    CTRP0760
C      *                                             2=Y COORD                    CTRP0770
C      *                                             3=Z COORD                    CTRP0780
C      *                                                                          CTRP0790
C      *       WRITTEN BY; LAEL R EASTERLING                                      CTRP0800
C      *       NPS, MONTEREY CA, FEBRUARY 1978                                    CTRP0810
C      *                                                                          CTRP0820
C      *                                                                          CTRP0830
C      ***************************************************************************CTRP0840
C      *                                                                          CTRP0850
C                                                                                 CTRP0860
C               ****************************                                      CTRP0870
C                                                                                 CTRP0880
```

```
C                   MAIN PROGRAM TO DETERMINE                              CTRP0890
C                   DIMENSIONS AND CALL SUB-                               CTRP0900
C                   ROUTINES                                               CTRP0910
C                                                                          CTRP0920
C                   *****************************                          CTRP0930
C                                                                          CTRP0940
      DIMENSION A(26000)                                                   CTRP0950
      COMMON/IPARA/NPCH                                                    CTRP0960
      MTCT=26000                                                           CTRP0970
      READ(5,1000)NN2,NPICT,NDMAX,NPCH                                     CTRP0980
      N1=1                                                                 CTRP0990
      N2=N1+NN2*2                                                          CTRP1000
      N3=N2+NN2*2                                                          CTRP1010
      N4=N3+NN2*2                                                          CTRP1020
      N5=N4+NN2*2                                                          CTRP1030
      N6=N5+NN2*2                                                          CTRP1040
      N7=N6+NN2*2                                                          CTRP1050
      N8=N7+NN2*2                                                          CTRP1060
      N9=N8+NDMAX*NPICT                                                    CTRP1070
      N10=N9+3*NN2                                                         CTRP1080
      N11=N10+NPICT                                                        CTRP1090
      N12=N11+3*NN2                                                        CTRP1100
      N13=N12+NPICT*20                                                     CTRP1110
      NLAST=N13+2*NPICT                                                    CTRP1120
      IF(NLAST.GT.MTOT)GO TO 111                                           CTRP1130
      CALL DEFPLT (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),              CTRP1140
     1A(N8),A(N9),A(N10),A(N11),A(N12),A(N13),NN2,NPICT,NDMAX)             CTRP1150
      STOP                                                                 CTRP1160
  111 WRITE(6,2000)NLAST                                                   CTRP1170
 2000 FORMAT(///1X,'NLAST=  ',I10,'MTCT INSUFFICIENT'///)                  CTRP1180
 1000 FORMAT(4I10)                                                         CTRP1190
      STOP                                                                 CTRP1200
      END                                                                  CTRP1210
C                                                                          CTRP1220
C                                                                          CTRP1230
C                         *************************                        CTRP1240
C                                                                          CTRP1250
C                         SUBROUTINE TO DEVELOP ANA-                       CTRP1260
C                         LYSIS PLANES AND CALL PLOT                       CTRP1270
C                         ROUTINE                                          CTRP1280
C                                                                          CTRP1290
C                         *************************                        CTRP1300
C                                                                          CTRP1310
      SUBROUTINE DEFPLT(XX,YY,ZZ,SIG1,SIG2,SIG3,SIGMX,LVL,COORD,KTR,       CTRP1320
     1PCOORD,TITLE,MC,NN2,NPICT,NDMAX)                                     CTRP1330
      REAL*8 XX(1),YY(1),ZZ(1),SIG1(1),SIG2(1),SIG3(1),SIGMX(1)            CTRP1340
      DIMENSION LVL(NPICT,1),COORD(NN2,1),KTR(1),PCOORD(NN2,1)             CTRP1350
      DIMENSION TITLE(NPICT,1),MC(2,1)                                     CTRP1360
```

```
      COMMON/IPARA/NPCH
      DO 10 I=1,NPICT
      KTR(I)=0
      DO 10 J=1,NDMAX
   10 LVL(I,J)=0
      REWIND 58
      READ(58)DUM
      DO 20 I=1,NN2
   20 READ(58)IDUM,XX(I),YY(I),ZZ(I)
      DO 35 I=1,NN2
      COORD(I,1)=XX(I)
      COORD(I,2)=YY(I)
      COORD(I,3)=ZZ(I)
   35 CONTINUE
      DO 30 I=1,NPICT
      READ(5,1002)(TITLE(I,JJ),JJ=1,20)
 1002 FORMAT(20A4)
      READ(5,1000)TESTL,TESTH,ICRD,IRD,ICT,MC(1,I),MC(2,I)
 1000 FORMAT(2F10.0,5I5)
      IF(IRD.EQ.1)GO TO 111
      DO 40 J=1,NN2
      IF(COORD(J,ICRD).GT.TESTL.AND.COORD(J,ICRD).LT.TESTH.OR.
     1COORD(J,ICRD).EQ.TESTL)GO TO 222
      GO TO 40
  222 KTR(I)=KTR(I)+1
      K=KTR(I)
      LVL(I,K)=J
   40 CONTINUE
      GO TO 30
  111 READ(5,1001)(LVL(I,L),L=1,ICT)
 1001 FORMAT(16I5)
      KTR(I)=ICT
   30 CONTINUE
      WRITE(6,2000)
      DO 45 I=1,NPICT
      K=KTR(I)
   45 WRITE(6,2001)I,(LVL(I,J),J=1,K)
      CALL MYPLOT(LVL,COORD,KTR,PCOORD,MC,NN2,NPICT,NDMAX)
      CALL PLTDT(SIG1,SIG2,SIG3,SIGMX,LVL,COORD,KTR,TITLE,MC,
     1NN2,NPICT)
 2000 FORMAT(//5X,'NODES NUMBERS BY ANALYSIS PLANE'///)
 2001 FORMAT(//5X,'PLANE NR ',I5//1X,4(1X,25I5/))
      RETURN
      END
C
C
C                  ************************************
C
```

```
C                    SUBROUTINE TC PLCT ANALYSIS PLANES                    CTRP1850
•                                                                          CTRP1860
C                   ***********************************                    CTRP1870
C                                                                          CTRP1880
                                                                           CTRP1890
      SLBROUTINE MYPLOT(LVL,COORD,KTR,PCCORD,MC,NN2,NPICT,NCMAX)           CTRP1900
      DIMENSION LVL(NPICT,1),KTR(1),CCCRD(NN2,1),FCCCRD(NN2,1)             CTRP1910
      DIMENSICN TITLE1(20)                                                 CTRP1920
      DIMENSION MC(2,1)                                                    CTRP1930
      CCMMON/IPARA/NPCH                                                    CTRP1940
      READ(5,1000)(TITLE1(I),I=1,20)                                       CTRP1950
      READ(5,1001)DMAGS                                                    CTRP1960
      CC 10 I=1,NN2                                                        CTRP1970
      CC 10 J=1,3                                                          CTRP1980
   10 PCCORD(I,J)=COORD(I,J)*DMAGS                                         CTRP1990
      CALL PLOTS                                                           CTRP2000
      CALL PLOT(-12.,3.,-3)                                                CTRP2010
      CALL PLOT(1.,0.,-3)                                                  CTRP2020
      CALL SYMBOL(0.,0.,.14,TITLE1,0.,8C)                                  CTRP2030
      CALL PLOT(4.25,5.25,-3)                                              CTRP2040
      CC 20 I=1,NPICT                                                      CTRP2050
      MX=MC(1,I)                                                           CTRP2060
      MY=MC(2,I)                                                           CTRP2070
      K=KTR(I)                                                             CTRP2080
      CC 30 J=1,K                                                          CTRP2090
      L=LVL(I,J)                                                           CTRP2100
      X=PCOORD(L,MX)-.02                                                   CTRP2110
      Y=PCOORD(L,MY)-.035                                                  CTRP2120
      CALL SYMBOL(X,Y,.C7,'X',0.,1)                                        CTRP2130
      X=X+.06                                                              CTRP2140
      Y=Y-.035                                                             CTRP2150
      F=FLOAT(L)                                                           CTRP2160
      CALL NUMBER(X,Y,.14,F,0.,-1)                                         CTRP2170
   30 CCNTINUE                                                             CTRP2180
      CALL PLCT(-4.25,5.,-3)                                               CTRP2190
      CALL SYMBOL(0.,0.,.14,'LEVEL',0.,5)                                  CTRP2200
      F=FLOAT(I)                                                           CTRP2210
      CALL NUMBER(1.0,0.,.14,F,0.,-1)                                      CTRP2220
      CALL PLCT(4.25,5.25,-3)                                              CTRP2230
   2C CCNTINUE                                                             CTRP2240
      CALL SYMBOL(-4.25,0.,.14,TITLE1,0.,19)                               CTRP2250
      CALL PLCT (-4.25,5.,-3)                                              CTRP2260
      CALL PLOTE                                                           CTRP2270
 1000 FCRMAT(20A4)                                                         CTRP2280
 1001 FCRMAT(F10.4,1I5)                                                    CTRP2290
      RETURN                                                               CTRP2300
      END                                                                  CTRP2310
C                                                                          CTRP2320
```

```
C                ************************************
C
C                SUBROUTINE TO PRINT ANC PUNCH CONTCUR
C                PLOT DATA
C
C                ************************************
C
      SUBROUTINE PLTDT(SIG1,SIG2,SIG3,SIGMX,LVL,CCORD,KTR,TITLE,MC,
     1NN2,NPICT)
      REAL*8 SIG1(NN2),SIG2(NN2),SIG3(NN2),SIGMX(NN2)
      DIMENSION LVL(NPICT,1),COORD(NN2,1),KTR(1)
      DIMENSICN TITLE(NPICT,1),MC(2,1)
      CCMMON/IPARA/NPCH
      REWIND 61
      READ(61)SIG1,SIG2,SIG3,SIGMX
      DO 10 I=1,NPICT
      MX=MC(1,I)
      MY=MC(2,I)
      IF(NPCH.EQ.0)WRITE(7,3002)(TITLE(I,JJ),JJ=1,20)
 3002 FORMAT(20A4)
      K=KTR(I)
      IF(NPCH.EQ.0)WRITE(7,3000)
      WRITE(6,2000)I
      WRITE(6,3003)(TITLE(I,JJ),JJ=1,20)
 3003 FCRMAT(//5X,20A4//)
      DC 20 J=1,K
      L=LVL(I,J)
      WRITE(6,2001)L,COCRD(L,MX),COORC(L,MY)
      IF(NPCH.EQ.0)WRITE(7,2002)L,COORD(L,MX),COORD(L,MY)
   20 CONTINUE
      IF(NPCH.EQ.0)WRITE(7,3001)
      WRITE(6,2003)
      DC 30 J=1,K
      L=LVL(I,J)
      WRITE(6,2004)L,SIGMX(L)
      IF(NPCH.EQ.0)WRITE(7,2005)L,SIGMX(L)
   30 CONTINUE
   10 CCNTINUE
 2000 FORMAT(//5X,'COORDINATES OF NODES TO BE USEC IN CONTCUR
     1PLOT--PLANE   ',I5//)
 2001 FCRMAT(1X,I5,2F15.5)
 2002 FCRMAT(I5,2F15.5)
 2003 FCRMAT(//5X,'STRESSES TO BE USED IN CONTOUR PLCT'//)
 2004 FCRMAT(1X,I5,F15.5)
 2005 FORMAT(I5,F15.5)
 3000 FCRMAT(6X,'FORMAT(I5,2F15.5)')
 3001 FORMAT('ENDGRID'/6X,'FORMAT(5I5)'/'ENDGEOM'/
     16X,'FORMAT(I5,F15.5)')
```

```
      RETURN                                                            CTRP2810
      END                                                               CTRP2820
```

LIST OF REFERENCES

1. Massachusetts Institute of Technology Report No. 82448-1, A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis (ADINA), by K. Bathe, May 1976.

2. Kibler, A. E., A. Finite Element Preprocessor for SAP IV and ADINA, M. S. Thesis, Naval Postgraduate School, Monterey, CA, September 1977.

3. Giles, G. L., Digital Computer Programs for Generating Oblique Orthographic Projections and Contour Plots, Document No. N75-17124, National Technical Information Service, U.S. Department of Commerce, January 1975.

4. AIRESEARCH Manufacturing Co. of Arizona, Blade Sections, Rotor, First Stage Turbine, Drawing No. D-TL-3103936, August 1976.

5. AIRESEARCH Manufacturing Co. of Arixona, Blade Turbine Rotor-First State Ceramic, Drawing No. E-99193-3101936, September 1976.

6. AIRESEARCH Manufacturing Co. of Arizona, Dovetail, Blade, Rotor, Drawing No. E-TL-965344, September 1976.

7. Preisel, J., Stress Analysis of Ceramic Turbine Blades by Finite Element Method - Part II, Engineer Thesis, Naval Postgraduate School, Monterey, CA, March 1978.

8. Bathe, K. J. and Wilson, E. L., Numerical Methods in Finite Element Analysis, 1st ed., Prentice Hall, 1976.

9. AIRESEARCH Manufacturing Co. of Arizona, First Design Review Meeting Presentation and Discussion of the Ceramic Gas Turbine Engine Demonstration Program, Report No. 76-211935-15, September 1976.

10. Pratt and Whitney Aircraft Division of United Technologies, Design, Fabrication and Spin Testing of Ceramic Blade-Metal Disk Attachment, Quarterly Report Nos. 1, 2, 3, and 4, Contract NAS3-19715, 1975-1976.

11. Anderson, C. A., Bratton, R. J., Cohn, A., Sanday, S. C. and Lange, F. F., Progress on Ceramic Rotor Blade Development for Industrial Gas Turbines, ASME paper 77-gt-42, Presented at ASME Gas Turbine Conference, 27-31 March 1977, Philadelphia, PA.

12. Naval Postgraduate School Technical Note No. 0141-03, Plotting Package for NPS IBM 360/67, by Patricia C. Johnson, August 1977.

13. Naval Postgraduate School Technical Note NO. 0141-28, Tektronix 4012 Computer Display Terminal with CP/CMS, User's Instruction Guide, by Richard Donat, May 1974.

14. Tektronix Inc., Plot-10 Terminal Control System User's Manual, Document No. 062-1438-00, May 1974.

15. Tektronix Inc., Plot-10 Advanced Graphing II User's Manual, Document No. 062-1530-00, 1973.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor Gilles Cantin, Code 69Ci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	12
5.	LCDR Lael R. Easterling, USN 4243 N. W. 54th Oklahoma City, Oklahoma 73112	2
6.	Georges Verchery Department de Genie Mecanique Universite de Technologie 60200 Compiegne, France	2
7.	Jean Louis Armand Institut de Recherche Pour la Construction Navale 3 Avenue de Grand Champ 78230 le Pecq, France	2
8.	Prof. K. J. Bathe Mechanical Engineering Department M. I. T. 77 Massachusetts Avenue Cambridge, Massachusetts 02139	2
9.	William J. Dodge Oak Ridge National Lab. Building 9204-1 Box Y Oak Ridge, Tennessee 36830	1

No. Copies

10.	Jack Tree Air Research Manufacturing Co. 402 South 36th Street P. O. Box 5217 Phoenix, Arizona 85010	2
11.	Prof. Edward L. Wilson Structural Engineering Division Civil Engineering Department University of California (Berkeley) Berkeley, California 94720	1
12.	Dr. William J. Stronge (Code 603) Naval Weapons Center China Lake, California 93955	2
13.	J. E. Serpanos Code 3162 Naval Weapons Center China Lake, California 93955	1
14.	Dr. Jean Louis Batoz, (I-230) Department of Civil Engineering Massachusetts Institute of Technology Cambridge, Massachusetts 02139	1
15.	Prof. Guri Dhatt Centre Technique de l'Informatique Universite Laval Quebec, Prov. de Quebec Canada, GIK7P4	1
16.	John Fairbanks Department of Energy Division of Power Systems 20 Massachusetts Ave. N. W. Washington, D. C. 20545	1
17.	Dr. Gilbert Tougot Centre d'Informatique Universite de Technologie 60206 Compiegne, France	1
18.	R. A. Langworthy Applied Technology Laboratories U. S. Army Research and Technology Laboratory Fort Eustis, Virginia 23604	1

No. Copies

19.	E. M. Lenoe Army Materials & Mechanic Research Center Arsenal Street Watertown, Massachusetts 02172	1
20.	Dr. Paris Genalis Naval Ship Research and Development Center Bethesda, Maryland 20084	1
21.	C. Miller Naval Sea Systems Command Department of the Navy Washington, D. C. 20362	1
22.	Code SEC 6734 Naval Ship Engineering Center Philadelphia Division Philadelphia, Pennsylvania 19112	1
23.	A. M. Diness (Code 471) Department of the Navy Office of Naval Research Arlington, Virginia 22217	1
24.	Ray M. Standahar Office of Secretary of Defense DDR&E 3D1089 Pentagon Washington, D. C. 20301	1
25.	R. Rice (Code 6360) Naval Research Laboratory Washington, D. C. 20375	1
26.	E. Van Reuth Defense Advanced Research Projects Agency 1440 Wilson Boulevard Arlington, Virginia 22209	1
27.	I. Machlin Naval Air System Command Department of the Navy Washington, D. C. 20361	1
28.	B. Probst, MS 49-3 NASA-Lewis Research Center 21000 Brookpark Road Cleveland, Ohio 44135	1

		No. Copies
29.	Mr. C. P. Blankenship, MS105-1 NASA-Lewis Research Center 21000 Brookpark Road Cleveland, Ohio 44135	1
30.	Dr. H. Graham/AFML/LLM Department of the Air Force Air Force Materials Laboratory Wright-Patterson Air Force Base, Ohio 45433	1
31.	Mr. George Strong DCASMA, Phoenix 3800 North Central Avenue Phoenix, Arizona 85012	1
32.	S. Freiman (Code 6363) Naval Research Laboratory Washington, D. C. 20375	1
33.	AFML/LLM/ N. M. Geyer Air Force Materials Laboratory Wright-Patterson Air Force Base, Ohio 45433	1
34.	S. Wiederhorn National Bureau of Standards Washington, D. C. 20234	1
35.	LT J. H. Preisel, Jr., USN 922 Bernard Road Peekskill, New York 10566	1
36.	Allan F. Greiner United Technologies Research Center East Hartford, Connecticut 06108	1

Thesis 175372
E1424 Easterling
c.1 Stress analysis of
ceramic turbine blades
by finite element
method - Part I.